JOACHIM BINGEL

PERSONALIZED AND ADAPTIVE TEXT SIMPLIFICATION

This thesis has been submitted to the PhD School of The Faculty of Science, University of Copenhagen.

PERSONALIZED AND ADAPTIVE TEXT SIMPLIFICATION

JOACHIM BINGEL

Ph.D. Thesis

August 2018

THESIS SUPERVISOR:
Anders Søgaard

ASSESSMENT COMMITTEE:
Christian Igel, University of Copenhagen
Andreas Vlachos, University of Cambridge
Katja Filippova, Google Research

AFFILIATION:
Department of Computer Science
Faculty of Science
University of Copenhagen

Joachim Bingel: *Personalized and Adaptive Text Simplification*, August 2018

ABSTRACT

Limited reading skills are a severe impediment for participation in our information-based society. Automatic text simplification has been suggested as an assistive technology to improve accessibility, but previous research has largely neglected variation between individual users and has suggested an objective notion of what makes text difficult and what does not.

However, as attested by previous research, readers perceive text difficulty individually and subjectively. Text simplification systems that assume general solutions and do not adjust to their individual users therefore cannot provide optimal solutions to the individual user, or by extension to the entire usership. Their potential is bound by the degree to which the target audience displays different simplification needs.

As a response, this thesis presents work that aims to integrate user information into the text simplification workflow, thus personalizing text simplification. This goal is pursued in two ways: (i) making it possible for users to state explicit simplification needs and preferences which the system, trained once on a static dataset, can then focus on at production time, and (ii) enabling a simplification model to learn from high-level user feedback and behavioral data in order to update its beliefs of a user's literacy profile. As an additional line of work, this thesis explores ways to build robust simplification models from limited training data, sharing information between smaller data sources through multi-task learning.

This work marks the first major effort to the development of text simplification systems that integrate information about individual users and adapt to their specific simplification needs. In personalizing text simplification, this user-focused technology can overcome existing upper bounds of performance and improve accessibility for weak readers.

ABSTRACT IN DANISH – RESUMÉ PÅ DANSK

Begrænsede læsefærdigheder er en alvorlig forhindring for at deltage i vores informationssamfund. Automatisk tekstsimplificering er et hjælpeværktøj der øger tilgængeligheden. Tidligere forskning har ikke taget individuelle hensyn til hvad der opleves som svært, men arbejder med objektive definitioner.

Men tidligere forskning viser, at sværhedsgraden af en tekst opleves individuelt og subjektivt. Tekstsimplificeringssystemer, som er underlagt antagelsen af én fælles løsning og ikke tager individuelle hensyn, leverer ikke den bedste løsning til den enkelte og dermed heller ikke til den samlede brugergruppe. Succesen af sådanne systemer afhænger af hvor forskellige målgruppens tekstsimplificeringsbehov er.

Derfor præsenterer denne afhandling arbejde hvor brugerinformation bliver integreret i tekstsimplificeringsmodellerne hvilket medfører individuel tekstsimplificering. Der er arbejdet med følgende løsninger: (i) at gøre det muligt for brugerne at give eksplicit udtryk for deres tekstsimplificeringsbehov og -præferencer som systemet, der ellers er trænet på det samme datasæt, kan tage udgangspunkt i, når systemet bruges og (ii) at anvende en simplificeringsmodel der lærer fra high-level brugerfeedback og brugerinput til at opdatere en model af en brugers læsefærdighedsprofil. I et andet spor undersøger denne afhandling måder at bygge robuste simplificeringsmodeller fra begrænsede mængder træningsdata ved at dele træningsdata fra flere mindre ressourcer igennem multi-task learning.

Dette arbejde bidrager til udviklingen af tekstsimplificeringssystemer som integrerer information om den enkelte bruger og tilpasser sig dennes simplificeringsbehov. Ved at individualisere tekstsimplificering er en brugertilpasset teknologi ikke underlagt en øvre resultatgrænse, som er betinget af forskelle mellem brugerne, og kan dermed øge tilgængeligheden for mennesker med læsevanskeligheder.

SUMMARY IN SIMPLE ENGLISH

People with weak reading skills have problems in finding jobs or reading important letters. Automatic text simplification can help to make text easier to read. For example, it can replace difficult words with simple words. But often, simplification programs do not know what is difficult for an individual.

This is a problem, because different people find different things difficult. If the program thinks that everybody has the same problems, it cannot work very well for a specific individual.

This book describes how text simplification can become better for specific individuals. For example, users can tell the program what they find very difficult. Then, the program can focus on these problems. This book also describes how the program can learn from the user. The user can say if they think the program is doing well. Then, the program can find out what the user finds difficult.

This is the first work to create simplification programs for individual readers. This can make it easier for many people to understand texts.

PUBLICATIONS

This thesis is built on the following peer-reviewed articles. They are identical in content as they appear here and in the original publications, with the exception of few minor changes such as the correction of typos.

Alva-Manchego, Fernando, Joachim Bingel, Gustavo Paetzold, Carolina Scarton, and Lucia Specia (2017). "Learning How to Simplify From Explicit Labeling of Complex-Simplified Text Pairs." In: *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Vol. 1, pp. 295–305.

Bingel, Joachim and Johannes Bjerva (2018). "Cross-lingual complex word identification with multitask learning." In: *Proceedings of the Complex Word Identification Shared Task at the 13th Workshop on Innovative Use of NLP for Building Educational Applications*. New Orleans, United States: Association for Computational Linguistics.

Bingel, Joachim and Anders Søgaard (2016). "Text simplification as tree labeling." In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Vol. 2, pp. 337–343.

Bingel, Joachim and Anders Søgaard (2017). "Identifying beneficial task relations for multi-task learning in deep neural networks." In: *15th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, pp. 164–169.

Bingel, Joachim, Gustavo Paetzold, and Anders Søgaard (2018a). "Lexi: a tool for adaptive, personalized text simplification." In: *COLING*, pp. 164–169.

Bingel, Joachim, Maria Barrett, and Sigrid Klerke (2018b). "Predicting misreadings from gaze in children with reading difficulties." In: *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pp. 24–34.

The following articles were also published along the way that led to the completion of this thesis. They did not make their way into the final product, but have in some way or another contributed to ideas reflected in the thesis.

Barrett, Maria, Joachim Bingel, Frank Keller, and Anders Søgaard (2016). "Weakly supervised part-of-speech tagging using eye-tracking data." In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. Vol. 2, pp. 579–584.

Barrett, Maria, Joachim Bingel, Nora Hollenstein, Marek Rei, and Anders Søgaard (2018). "Sequence classification with human attention." In: *Proceedings of the 22nd Conference on Computational Natural Language Learning*.

Bingel, Joachim, Natalie Schluter, and Héctor Martínez Alonso (2016a). "CoastalCPH at SemEval-2016 Task 11: The importance of designing your Neural Networks right." In: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pp. 1028–1033.

Bingel, Joachim, Maria Barrett, and Anders Søgaard (2016b). "Extracting token-level signals of syntactic processing from fMRI-with an application to PoS induction." In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. Vol. 1, pp. 747–755.

Bollmann, Marcel, Joachim Bingel, and Anders Søgaard (2017). "Learning attention for historical text normalization by learning to pronounce." In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*. Vol. 1, pp. 332–344.

Bollmann, Marcel, Anders Søgaard, and Joachim Bingel (2018). "Multitask learning for historical text normalization: Size matters." In: *Proceedings of the Workshop on Deep Learning Approaches for Low-Resource NLP*, pp. 19–24.

Ruder, Sebastian, Joachim Bingel, Isabelle Augenstein, and Anders Søgaard (2017). "Learning what to share between loosely related tasks." In: *arXiv preprint arXiv:1705.08142*.

Waseem Butt, Zeerak, James Thorne, and Joachim Bingel (2018). "Bridging the Gaps: Multi-Task Learning for Domain Transfer of Hate Speech Detection." In: *Online Harrassment*. Ed. by Jennifer Goldbeck. London: Springer.

Taking things for granted is a terrible disease.
We should all be checking ourselves regularly for signs of it.

— Kate Tempest, Hold Your Own

ACKNOWLEDGMENTS

I would like to express my gratitude to a number of people who in some way or another, during this journey, were at my side.

The journey began in September 2015 when I stepped into the corridor where the CoAStaL research group had its offices. From the first moment, my colleagues made me feel a part of the group. Anders J., Barbara, Héctor, Natalie, Željko – thank you so much! Maria and Sigrid, you deserve a special 'thank you' for taking me by the hand and making the trip so much easier, not only in the beginning. Dirk, I can't stress enough what a great colleague you have been. For being the best imaginable co-teacher, a most helpful adviser in academic questions of any kind, an incredible conference buddy, and for never letting me include an outline slide in my presentations, thank you! Dear first-hour colleagues, you're wonderful people, and all of you have been great role models. I have been fortunate to learn so much from you.

Needless to say, I also owe gratitude to you, Anders. On this journey, you've been exactly the supervisor I needed and perhaps even a better one than I could have wished for, although perhaps *vejleder* is the better word. Thank you for your ever-open door, unconditional support, for creating a wonderful research group, for endless optimism, an inexhaustible spring of ideas and for leading by example.

Many more colleagues later joined along the way. In mentally reconstructed order of appearance, I'd like to thank Andreas, Chloé, Mareike, Ophélie, Isabelle, Johannes, Yova, Marcel, Meriem, Ana and Simon for good times by the coffee machine, at Caféen or at conferences near and far. During the three months I spent at the University of Sheffield, other people came into the picture that I would like to thank just as much. Andreas, Carol, Chiraag, Fernando, Fred, Gustavo, James, Johann, Josiah, Lucia, Makis, Pranava and of course Zeerak – thank you all for taking good care of me.

I have been fortunate to see several aspects of my work find interest outside the academic ivory tower. For exciting collaborations and their embracing of academic peculiarities, I want to thank Janus and Emil from EyeJustRead as well as the people at Nota, most notably Bolette and Sara. The NLP team at Amazon Research Berlin hosted me for three months in 2017, and I am grateful in particular to Weiwei and Alex for being so welcoming.

Acknowledgement sections often include the phrase “it wouldn’t have been possible without”. Being extra enthusiastic in the acknowledgements is probably okay, but I honestly cannot imagine how much harder my work would have been if it hadn’t been for the incredible effort and dedication that thousands of individuals put into the development and maintenance of free and open source software packages during their free time. Thank you so much!

Another “it wouldn’t have been possible without” goes to an even wider circle of people: From the first day in kindergarten to the day I handed in my PhD thesis, I spent a net time of roughly 26 years in the educational systems of Germany, Sweden and Denmark, plus a short stunt in the UK. All of this was free of charge. I strongly believe this is how it should be, but I also know I’ve been incredibly privileged to benefit from this free education. Dear taxpayers, thank you sincerely. I look forward to further returning the favor. Schools and universities are notoriously underfunded, so let’s spend more on education, at home and globally, and many problems will become a lot smaller.

To all the people I’m lucky to call my friends, I am more than grateful for all your support before, during and after the PhD years. I’m really fortunate that you give me so many good distractions from my work and that you’ve never really given me a chance to become a workaholic. I’ve moved around a bit, but wherever I’ve been you’ve been making me feel at home.

I owe thanks to my family, who has been with me from the very beginning and always will be. Danke Mama, Papa und Biggi, Chris, Marius, Lotta und Luci, meinen Großeltern, und allen weiteren, i també vull dir gràcies a la meva familia catalana. Ihr seid schon so viele Wege mit mir und für mich gegangen, und habt mir die Unterstützung gegeben, ohne die ich vieles nicht geschafft hätte.

Finally, my deepest gratitude goes to Joana. Living in different countries for three years was hard at times, but more than anyone you’ve been walking by my side. I won’t find the words that will do justice to the gratefulness that I feel and that you deserve, but for your patience during this time, your affection, for overcoming the physical distance between us in more than one way, and for challenging me on and off the academic pitch, you have my endless thankfulness and love.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Part I

BACKGROUND

1 INTRODUCTION

1.1 LITERACY AND ACCESSIBILITY

About 5,000 years ago, somewhere in ancient Mesopotamia, humans invented a system of documenting, preserving and transmitting information in the form of script. Writing enabled people to encode information they naturally transmit in speech in a permanent medium. Conversely, this information could now be decoded through the act of reading, that is the translation of abstract symbols engraved into stone or drawn on papyrus or canvas into any sort of information they would normally receive through auditory means. These 5,000 years of reading and writing have massively transformed many aspects of human culture and civilization. Relatively recent developments such as globalization and digitization, in particular, would hardly or not at all be possible without our ability to store, multiply, manipulate and receive information we would otherwise only be able to communicate transiently and only within our immediate radius.

From the perspective of human evolution, however, literacy is a very recent development that has hardly had any time to manifest itself in our biological legacy. If 5,000 years seems like a short time for genetic change, the period in which it has been a societal – and ultimately evolutionary – advantage to read and write hardly exceeds a couple of generations in most cultures. And even in those where it does, only a small fraction of the population has been literate for more than 200 years (Buringh and Van Zanden, 2009). Reading has been a cultural technique historically commanded by select members of a society, rather than a survival strategy necessary for everyone.

Consequently, reading is a skill that is nowhere near innateness. Instead, in order to acquire it, most of us train almost daily for a period of several years at an age where we are most susceptible to new intellectual challenges. Yet, despite the efforts many governments and societies have invested in literacy training via public schooling over the last centuries, many learners never make it past basic reading skills. In many cases, this is not due to bad or insufficient teaching, or to a lack of motivation and dedication on the learner's part, but to neurological conditions such as dyslexia, aphasia, autism and others. Dyslexia, in particular, is a highly prevalent learning disability that manifests as an increased difficulty in reading, especially learning the mapping between sounds and letters and decoding written material into otherwise familiar words, despite normal intelligence. Consequently, people with dyslexia experience difficulties in fluent and accurate reading and text comprehension as well as spelling. Dyslexia can be present from birth or develop through environmental factors (Peterson and Pennington, 2015). Estimates of its prevalence differ widely and are subject to different definitions, but generally range between 5 and 18 per cent of English native speakers (Peterson and Pennington, 2015; Interagency Committee on Learning Disabilities, 1987; Shaywitz et al., 1992; Katusic et al., 2001). For other languages, figures are similar, yet slightly lower for languages with a more direct grapheme-to-phoneme correspondence, such as Italian or Spanish (Brunswick, 2010; Carrillo et al., 2011).

While research is ongoing to understand and fight cognitive challenges such as dyslexia, policy makers and private initiatives have been implementing policies that promote accessibility to written sources for low-literacy readers – who comprise, besides people with learning difficulties, a number of other groups such as second-language learners and beginner readers. These policies can most notably be found in the form of easy language guidelines, which are followed by different actors to provide easier access to their materials. Examples include easy-read editions of authorities' websites and leaflets, special issues of newswire texts, or literature. [1]

The importance of these efforts can hardly be overstated. In present-day society, literacy has become an indispensable skill. While reading or writing may not be directly necessary for survival, social security and economic independence certainly depend on them – not to mention higher-level human needs such as societal recognition, and feelings of accomplishment and self-fulfillment. It is therefore vital to further promote accessibility and the inclusion of low-literacy readers into our knowledge-based society.

This forms the primary motivation for automatic text simplification, which aims at providing simple versions to some given input text on demand and within a negligible amount of processing time, without the need for an expensive and, above all, slow human expert. The case for simplification and how it relates to accessibility is explicated further by Siddharthan (2014): difficulty in word recognition entails difficulty in higher level processing due to the overly high occupation of the working memory (Anderson, 1981; Quigley and Paul, 1984), such that any effort that facilitates the former can be expected to lead to overall improvements in reading comprehension (Mason and Kendall, 1979). This thesis presents and discusses research in automatic text simplification carried out by the author

[1] For an example of an authority website in simple language, see the German government's website, which offers much of its content in Simple German: https://www.bundesregierung.de. Examples of news platforms in simple language include The Times (http://www.thetimesinplainenglish.com/), German taz (http://www.taz.de/!5425192/) and Danish DR Ligetil (https://www.dr.dk/ligetil). A book publisher specializing in simple language is the Swedish LL-förlaget (http://ll-forlaget.se)

over the course of three years. The focus, as will be established in the remainder of this chapter, is on the development of adaptive and personalized frameworks and models for text simplification. This is in response to the observation that generic approaches, which assume homogeneous target populations with consistent simplification needs, are not generally adequate, given the highly individual perceptions of word or text difficulty experienced by different end users.

1.2 TEXT SIMPLIFICATION: THE BASICS

A common view of text simplification as a research direction in Natural Language Processing (NLP) is rewriting text such that it becomes easier to read. In fact, many research papers dedicated to simplification begin with this basic definition and, depending on their focus, include a number of examples that showcase ways in which text can be simplified (Zhu et al., 2010; Coster and Kauchak, 2011c; De Belder and Moens, 2010; Paetzold and Specia, 2015; Bingel and Søgaard, 2016). For example, simplifying texts may involve the substitution of difficult words with easier ones, shortening sentences by removing peripheral information or splitting them up into several shorter ones, or replacing pronouns with their referents, among others. The following examples, all taken from the professionally edited and simplified Newsela corpus (Xu et al., 2015), showcase different simplification strategies, contrasting original sentences with their simplified counterparts. Consider the first example:

(1) a. Even as wolverines rebound, threats loom in their future.

b. Even as wolverines return, they still face threats.

Here, the word *rebound* has been replaced by *return* in the simplified version. This is an instance of lexical simplification, where the syntactic and semantic structure of a sentence or a clause is retained, but individual words or phrases are exchanged for simpler synonyms. The part of the original sentence following the comma is slightly more complex. It is paraphrased as a whole, with the metaphor of the *looming threats* being translated into a more literal expression and a change of focus from the abstract threats to the wolverines. A similar instance is found in example 2, where an entire phrase is replaced with an alternative.

(2) a. European leaders cannot agree on how to handle the problems facing their continent.

b. European leaders are not sure how to handle these problems.

This example highlights a common difficulty in simplification: as in this case, paraphrases are never perfect synonyms – not being able to agree on something is not quite the same as not being sure about it. In general, supposedly synonymous terms, even when they are equal or almost equal in their denotation, usually differ at least to some degree along other dimensions such as formality or other semantic and pragmatic connotations (Stanojević, 2009).

Another simplification strategy present in example 2 is the omission of secondary or peripheral sentence material, often in the form of modifiers such as adverbials. Again, this may entail a more or less severe distortion or loss of information, such that simplification efforts are faced with the challenge of achieving simplicity while preserving meaning.

The next example presents a case that rewrites text material of informal style, which is not generally encountered in text books or other material consumed by most groups usually targeted in simplification, in particular second-language learners. The simplification strategy employed in this example is therefore to rewrite the informal bits of text in a way that is closer to the standard form.

(3) a. Anxiety doesn’t cause the yips, crews say, but it can make the problem worse.

b. Crews say being nervous can make the problem worse.

In example 3, we also note that the editor resolves the interjected subject and predicate and moves them to the front of the sentence, making it easier for the reader to interpret the reported speech as such. We observe the same simplification strategy in example 4. It further removes largely peripheral material found in the final clause of the original sentence, whose pragmatic function supposedly is of somewhat literary character and therefore secondary in relation to the main assertion.

(4) a. “This kind of sport is not only popular in Xinjiang , it’s also pretty popular across China," Jin said, taking a rest from one of his workouts on a Sunday afternoon.

b. Jin said the sport is popular across China.

In automatic text simplification, the central challenges we face revolve around a range of questions on how to perform these edits. How can we detect material in a text that is particularly difficult for readers? How can we automatically generate alternatives to such material? How can we decide which of the alternatives both fit the context and make reading as easy as possible – or as easy as desired? These questions are not generally easy to answer. Early stages of text simplification research have aimed to build rule-based systems that treat very specific grammatical phenomena and simplify these. Striving for more robustness and coverage, most research has for the last decades tried to induce simplification systems from data using machine learning techniques. A subset of these efforts has focused on specific simplification strategies such as lexical simplification or sentence compression, while others have attempted to generate more holistic models that simplify a given text on a number of levels. The survey of text simplification research presented in Chapter 2 goes into more detail.

A much more basic question in simplification research, however, touches the definition of the task itself. The examples above demonstrate simplification strategies that correspond to the basic definition of the task given at the beginning of this section: given a piece of text, rewrite it such that it becomes easier to read. Here, no mention is made of any specific target audiences, just as none of the examples explicitly states which parts of the original sentence should be modified in an effort to accommodate the simplification needs of a particular target group or individual.

There are, however, a number of research efforts in text simplification that extend the above definition and explicitly address certain target audiences, such as foreign language learners or people with learning disabilities. Some of these are discussed in greater detail in the next chapter. Before we turn to that, the following section argues why the classical understanding of text simplification is not sufficient and sets the theoretical scene for the remainder of the thesis.

1.3 there is no one-size-fits-all solution

The definition of text simplification given above is straightforward and arguably frames the essence of the task reasonably well. However, as we will argue in this section and in other parts of this dissertation, it lacks the critical dimension of personalization, i.e., a necessary consideration of the specific simplification needs of an individual user. Without this dimension, and without an embedding in real-world situations, text simplification becomes a lot less valuable outside the academic ivory tower.

Text simplification differs from many other tasks in NLP in that it is not always easy to decide whether the output generated by some model is good. In part-of-speech tagging or parsing, even though there may be competing theories or structural ambiguities, we can typically judge the quality of a prediction objectively and with high confidence.[2] The same goes for other typical NLP task families such as information extraction or text classification. Machine translation is closer to simplification in this respect: while some translations may be more canonical than others, we can also generally accept many different hypotheses. However, questions of style and linguistic com-

[2] Furthermore, these kinds of grammatical analysis are typically employed as first steps in a pipeline and not directly relevant to the end user.

plexity can generally be answered relatively objectively, given that a translation should reflect these qualities as they appear in the source.

In the case of text simplification, this is different. Here, at least certain aspects of the predicted output of a model must be judged by the end user, subjectively and with respect to their specific simplification needs. The reason for this is obvious: as an assistive technology, text simplification serves the central purpose of promoting accessibility of written language for people who would otherwise not be able to understand it fully or be able to do so only to some degree, or who would have to invest excessive amounts of energy to do so. In conjunction with the fact that simplification needs and subjective notions of text difficulty are highly specific not only between groups of people (e.g. dyslexics vs. foreign language learners), but also intra-group and between individuals, this means that simplification efforts must be developed and evaluated with respect to individuals.

Evidence for the highly individual perception of text difficulty in dyslexics, for instance, is provided by Watson and Goldgar (1988), Bakker (1992), and Ziegler et al. (2008). People on the autism spectrum, who have been addressed widely in readability and simplification research (Yaneva, 2016; Yaneva et al., 2016a,b), have been found to exhibit very different manifestations of their condition in general, but also with respect to reading (Evans et al., 2014). The need for truly individualized approaches to simplification is further supported by the fact that while researchers agree that some typologies of dyslexia or autism exist, the particular typologies that have been proposed are hotly debated. This makes it less feasible to develop simplification solutions for specific subtypes of these conditions.

A number of other groups that have been targeted by simplification display very individual cases that require personalized simplification strategies. In the case of second language acquisition, a learner's native language unsurprisingly has a profound influence on difficulty judgements (Shatz, 2017), but obviously this also varies strongly within a group depending on factors such as the knowledge of other foreign languages or simply dedication and talent. Beginner readers evidently are very individual in the development of their literacy, progressing at an individual pace and reaching different reading levels even within the same class and age.

Very recently, first empirical evidence of the aptitude of personalized simplification models has been presented by Yimam and Biemann (2018) and Lee and Yeung (2018). Both of these works acknowledge that individual differences between users demand personalized solutions to the simplification task and explore ways to adapt simplification models to individual needs. The experiments by Yimam and Biemann (2018) show that re-training a substitution ranking model on progressively more ranking annotations from the same user lets the model adapt to this user quickly and effectively, with an error reduc-

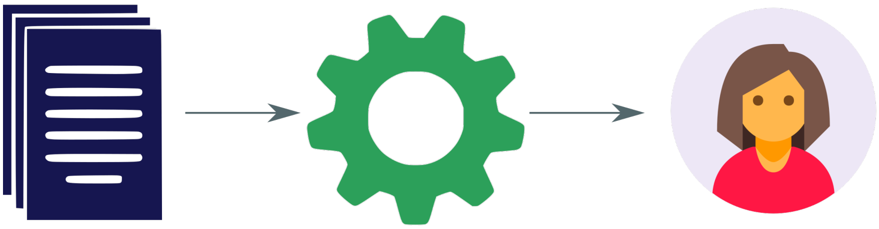


Figure 1.1: Traditional simplification workflow. From a static dataset, a generic model is induced, and the user is presented with whatever the model outputs, with no opportunity for the user to control model behavior.

tion of over 30% after rating items in just over 1,000 sentences. Similarly, Lee and Yeung (2018) conduct experiments in complex word identification, a first step in lexical simplification, that show the superiority of models trained on individual annotations over generic ones.

The above observations form the central motivation for the work carried out in the framework of this thesis: there is no generic, one-size-fits-all solution to text simplification. Instead, in order to truly be helpful to the end user, it needs to be personalized or personalizable.

1.4 BEYOND END-TO-END LEARNING IN SIMPLIFICATION

As a sub-discipline in NLP, simplification faces the challenge of dealing with natural language. Its organic nature makes language extremely difficult to describe in all its subtleties and variations. Anybody who has ever ventured to learn a foreign language has experienced the frustration that many phenomena cannot easily be described with, let alone be explained by, a manageable set of rules. Yet even where rules are known, expressing them in unambiguous logical forms such that computer programs can deal with them is often an intractable problem.

Therefore, language technology today largely pursues approaches that are based on statistical methods, in particular machine learning techniques. In basic terms, machine learning techniques aim to induce a mathematical model from observed data in order to make decisions on new data observed at a later time. In the context of text simplification, this may involve training a translation-like model on pairs of "normal" and simplified sentences, or learning to detect difficult words from annotations of language learners. In the optimal case, the model learns to reproduce exactly what it sees in the data it is trained on, while still generalizing optimally to new, unseen data at test time. This process is illustrated in Figure 1.1, where a model is induced from data in order to later provide simplifications to a user. In this scheme, the model is entirely ignorant of the user, such that its output is determined solely by the data it was trained on and the input it receives at production time, but not by any user-dependent variables. More formally, the output y is defined as

$$y = M_D(x) \tag{1.1}$$

where M denotes the model induced from some dataset D, and x is the production-time input. Generally, full generalizational power is not easily attained – although, even if it were, models trained end-to-end on a particular dataset will always only be as good as that dataset, with no way of adapting to the specific simplification needs of an end user. Moreover, supervised learning from a static dataset introduces inherent biases towards the text domains and language varieties that are covered by it, such that the successful application of a statically trained system is hampered when the user's reading interests differ from those domains or when the target domains themselves change over time (Žliobaitė et al., 2016).

The present dissertation aims to overcome these limitations. Rather than focusing on developing state-of-the-art models on some benchmark dataset, its central research question is:

How can we make simplification more adaptive
and personalizable to the individual end user?

This is tackled mainly along two dimensions: (i) the development of parametric simplification methods that allow for the setting of user preferences at production time, as well as (ii) the incorporation of user-specific behavioral data into the simplification model in order to foster the continuous optimization and personalization of the model. A third line of work tackled in this dissertation addresses generalization of simplification models across different settings, including different text domains but also languages, using multi-task learning.

Eventually, the aim is to devise models that generate output depending upon user preferences and some form of feedback. This is reflected in the scheme depicted in Figure 1.2, which incorporates a feedback loop from the user to the system, but also an interface to explicitly set particular preferences for the user. This extends Equation 1.1 to

$$y = M_{D,H}(x, \pi) \tag{1.2}$$

where the model M is now initially induced from a base dataset D and then continuously updated in an online fashion from a history H of user feedback. The output at production time is then conditioned on the input x as well as explicitly set preferences π. Besides adapting a model to a specific user, the continuous integration of user feedback generally reduces offline annotation effort (To et al., 2009) and thus also reduces the problem of resource scarcity that we typically face in text simplification and other application areas of natural language processing (see section 2.2.3 for a discussion of problems with resource scarcity and possible mitigations).

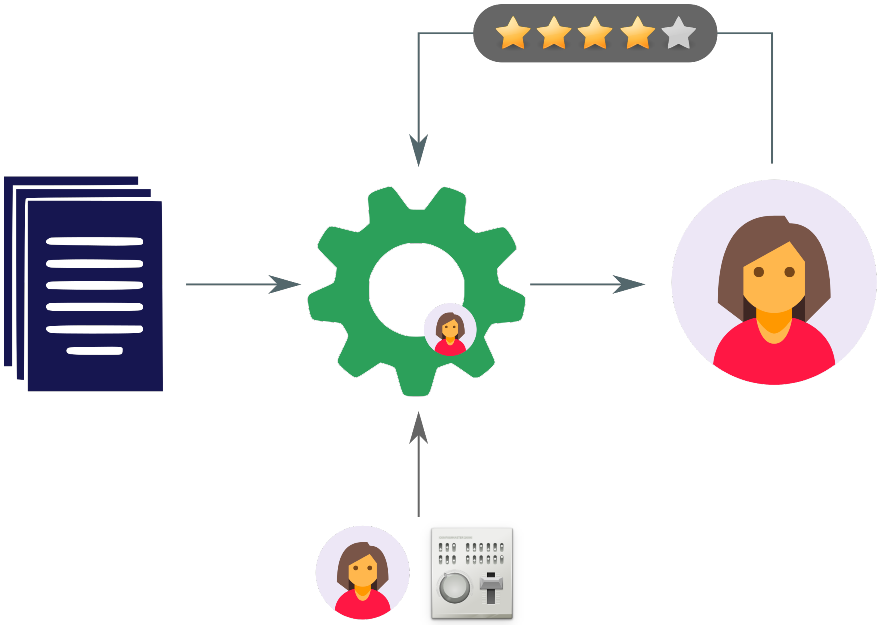


Figure 1.2: Extended simplification workflow incorporating user preferences and feedback. The user-specific model accepts the setting of user preferences and conditions the output on this. It further updates itself continuously from user feedback.

1.5 MAIN CONTRIBUTIONS

At this point, let us review the definition of text simplification as stated initially in Section 1.2, which portrays the task in a largely user-agnostic fashion. The foregoing discussion motivates an extension of that definition to include knowledge of the end user in the simplification process, with the aim of tailoring all simplifications as appropriately as possible to the specific user. In other words, let us consider text simplification as a task that aims to make text easier to read *for a specific person, according to what is known about their specific simplification needs and reading proficiency*, while preserving as much as possible of the informational content.

The individual studies carried out in the framework of this thesis are, in some way or another, reflections of this updated definition of the simplification task. They address various aspects of an extended, user-aware conception of text simplification and thus constitute the main contributions of this thesis:

- Chapters 3 and 4 introduce and evaluate methods to induce sentence-level simplification models that allow for live adaptations at production time. This is achieved through overcoming the black-box character that typical end-to-end simplification methods entail, and instead explicitly predicting available simplification operations for various parts of a sentence. These can then be used by parametric models that possess knowledge of their users to generate custom-tailored sentence simplifications.

- Besides explicitly setting user preferences (such as a hard limit on sentence length or a rule that all pronouns should be replaced with their antecedents), we would like to infer those more naturally from various kinds of user feedback. This is motivated by the assumption that individuals do not generally have detailed insights into their particular simplification needs, e.g., the particular factors that render a word difficult to read for them. Methods to learn user preferences from explicit and implicit feedback such as behavioral data are proposed and discussed in chapters 5 and 6.

- Standard machine learning methods that induce a model from a single dataset are prone to overfitting and limited adaptiveness to new domains or other forms of divergence of data distributions at training and testing time. This can be alleviated by multi-task learning, i.e., the learning of various functions in a single model, possibly from disparate datasets. Multi-task learning models are discussed in Chapters 7 and 8, shedding light on the conditions under which the paradigm is helpful and how it can help to transfer knowledge between tasks.

2 A SHORT HISTORY OF SIMPLIFICATION RESEARCH

Text simplification as a research direction in NLP dates back to the 1990s and has its origins in the works of Chandrasekar et al. (1996), Chandrasekar and Srinivas (1997), and Dras (1999). Interestingly, making text more accessible for *human* readers is *not* the first devised application for simplification as considered in these papers. Instead, the case for simplification derives from the observation that long sentences with uncanonical formulations and low-frequency vocabulary will be particularly difficult to process for machines, for instance in NLP application areas such as parsing, machine translation and information extraction.[3]

Nevertheless, these early works established text simplification as a research direction and inspired follow-up studies that approached the task as an assistive technology directed at human users. Among the first of these is the work by Carroll et al. (1998), which puts its focus on a specific target group, tackling problems that aphasic readers typically face. This is remarkable, because subsequent efforts in text simplification have often neglected any specific simplification needs of particular groups, and have instead assumed a rather homogeneous user population.

Another development concerns the actual methods that have been applied in creating simplification systems. The earliest works by Chandrasekar et al. are based on explicit syntactic transformation rules that are either hand-coded or induced from data. As argued by Siddharthan (2014), hand-crafted rules are sensible in the realm of text simplification when a system focuses on very specific linguistic structures and phenomena that are relatively easy to manage with a finite and small set of rules. However, such rule-based systems often suffer from very limited coverage and fail to detect subtle variations in surface form. A focus on the simplification of lexical material further requires some sort of data-driven approach.

At a very coarse level, a common distinction between research efforts in text simplification divides them into those focusing on lexical simplification, i.e., the replacement of individual words or multiword expressions with simpler alternatives, and those that take a more holistic approach and aim to rewrite entire phrases or sentences. In the latter case, typical simplification strategies are to shorten sentences by removing peripheral information, to break up a long sen-

[3] In the light of this observation, the lack of user orientation discussed in the previous chapter and in further parts of this thesis may appear less surprising.

tence into several shorter ones, or to transform passive constructions into active ones – see the previous chapter for a range of sample simplification strategies.

Notable, extensive surveys on text simplification have been published by Siddharthan (2014) and Shardlow (2014a), as well as by Paetzold and Specia (2017a) with a focus on lexical simplification. The overview presented here draws on these surveys and complements them with recent developments and, where applicable, an angle on the target audiences of individual works and the question of how adaptable these works are. Following the conceptual distinction between lexical and more holistic, sentence-level simplification, we first give an account of previous research in the former field, then in the latter, in the following two sections. We further discuss the particular challenges and solutions in both of these areas.

2.1 LEXICAL SIMPLIFICATION

Sometimes overlooked in related work sections, dedicated lexical simplification components have been part of higher-level text simplification and adaptation systems since Carroll et al. (1998), who base their work on that of Devlin and Tait (1998). The latter employs a simple synonym lookup in WordNet (Miller, 1998) as well as the Oxford Psycholinguistic Database (Quinlan, 1992), which encodes word difficulty through the Kučera-Francis frequency, i.e., the relative frequency of a word in a million-word corpus (Kučera and Francis, 1967; Rudell, 1993). Interestingly, this work also includes a way for users to control the output by specifying a desired level of simplification. Very similar approaches to lexical simplification, yet without any control through the user, are offered by Lal and Ruger (2002), Burstein et al. (2007), and De Belder et al. (2010).

2.1.1 *Data-driven approaches*

However, lexical simplification only gained traction as a more focused research direction with the work of Yatskar et al. (2010). Their approach overcomes the limitations of previous lexical simplification approaches, which could only retrieve candidate replacements from precompiled synonym dictionaries. Instead, Yatskar et al. (2010) exploit edit histories in the Simple English Wikipedia[4] and mine synonym pairs from the old and updated versions of articles, calculating the probabilitiy that an edit is in fact a simplification operation. This contribution inspired a host of papers tackling lexical simplification; Biran et al. (2011), for example, compare the simple and "regular" English Wikipedia texts and extract simplification rules from statistics of word distributions over these.

[4] https://simple.wikipedia.org

Further developments then saw the Lexical Simplification shared task at SemEval 2012 (Specia et al., 2012). This task featured simplicity annotations from non-native speakers of English and focused on the ranking of a set of candidate replacements for a given word in context. Receiving submissions from five teams, the task was "a first attempt at garnering interest in the NLP community for research focused on the lexical aspects of Text Simplification" (Specia et al., 2012), and did indeed help place this research direction on a sounder footing. The findings of the shared task included, among others, further evidence of the strong relationship between frequency and simplicity, at least in the case of the dataset used in the task.

Lexical simplification was later tackled in a concentrated research program by Matthew Shardlow. One of his contributions is the definition and formulation of what is known as the Lexical Simplification Pipeline, a four-step process that (i) identifies difficult material in a sentence, (ii) generates candidate replacements for these, (iii) filters out candidates not fitting the context, and (iv) ranks the remaining substitution candidates by simplicity (Shardlow, 2014b). This procedure, illustrated in Figure 6.1, has become a *de-facto* standard conceptualization of the task and has been adopted by many subsequent works.

In particular, this division into several subtasks has led to concentrated efforts on individual aspects of the pipeline, most notably complex word identification (CWI) and substitution ranking (SR). Two shared tasks with a combined number of 33 research teams have been held for complex word identification (Paetzold and Specia, 2016b; Yimam et al., 2018), with the winning systems typically employing ensembles of systems that make use of a number of manually engineered features of linguistic and psycholinguistic nature. Both shared tasks provide manual annotations of word difficulty. In the case of Paetzold and Specia (2016b), these come exclusively from second language learners of English, whereas Yimam et al. (2018) include annotations by native speakers and extend the task to three additional languages (German, Spanish and French). The second shared task also differentiates itself from its predecessor through its "probabilistic" track, i.e., the challenge to predict *the fraction* of annotators who deemed an item difficult, as opposed to the binary decision whether *any* of the 10 or 20 annotators did so. From the perspective of this thesis, this gradual notion of difficulty is a lot more desirable than a binary one, as it allows for much simpler adaptivity in the lexical simplification pipeline, for instance via a threshold that can express the level of simplification a user desires. However, the 2018 shared task was not limited to the probabilistic setting and additionally offered a binary track, which received a lot more attention from the participants and still seems to be favored by the community.

Substitution ranking was approached in a very simple manner in early lexical simplification research, mostly by taking frequency as a proxy for simplicity (Devlin and Tait, 1998; Carroll et al., 1998; Shardlow, 2014b). While frequency is indeed a very strong signal of expert simplicity ratings (Rudell, 1993), the approach has a number of shortcomings. One of these is that it does not generalize well across target audiences: frequency correlates with (and causes) familiarity, which may be beneficial for second language learners with good general reading skills. However, other properties of a written word, such as its length or the occurrence of certain character combinations, may be much more relevant to the difficulty perception in dyslexics or other populations with generally low literacy. Research in substitution ranking has therefore included further word representations through features such as length or abstractness (Biran et al., 2011; Sinha, 2012). These and a range of following approaches have in common that they compute simplicity scores independently across a set of substitution candidates. As such, they do not technically differ from the CWI approaches listed above in a significant way. Work that addresses the *pairwise* ranking of candidates is presented by Paetzold and Specia (2017b), which is also one of the first successful applications of deep learning in lexical simplification and uses a Siamese neural network to decide the relative difficulty between two words. This has been shown to be superior to more classical approaches and currently represents the state of the art.

Yimam and Biemann (2018) as well as Lee and Yeung (2018) recently published first work on personalized and adaptive lexical simplification, demonstrating that models trained on user-specific data, even when small, work better for the same user than generic models. They focus on substitution ranking and complex word identification, respectively, and Yimam and Biemann in particular demonstrate the effectiveness of an adaptive setup, where the system learns from continuous user feedback.

2.1.2 *Languages other than English*

Depending upon the availability of lexicographical resources on the one side and parallel corpora on the other, data-driven approaches can be a way to develop lexical simplification systems for languages other than English. Lexical simplification for languages other than English comprises the efforts of Bott et al. (2012a) and Drndarevic and Saggion (2012), who have been focusing on Spanish lexical simplification. Ferrés et al. (2017) present a multilingual approach that works across the major Ibero-Romance languages (Spanish, Portuguese, Catalan and Galician). As an example of lexical simplification outside the Indo-European family, Kajiwara and Yamamoto (2015) and Hading et al. (2016) present work on Japanese.

2.1.3 *Open challenges in lexical simplification*

As outlined above, lexical simplification has not received an overwhelming amount of attention from the research community. Consequently, while the task may at first sight seem relatively simple, there are a number of challenges that require further research into this area. The major ones are discussed in this section.

NEAR-SYNONYMS In substitution generation, i.e., the retrieval of candidate replacements for some target word, the most straightforward approach is to use machine-readable synonym dictionaries or a WordNet (Miller, 1998) to look up synonymous expressions for a target word. This solution typically offers robust synonym relations edited by professional lexicographers. However, such dictionaries may suffer from low coverage or, for most languages, be entirely unavailable. For less resourced languages, a common approach is thus to retrieve synonyms from corpus statistics. Departing from the assumption that words similar in meaning occur in similar linguistic environments, word embedding algorithms compute high-dimensional, real-valued vector representations of word meaning, such that words with similar meanings are clustered in close proximity within the vector space (Bojanowski et al., 2016; Mikolov et al., 2013). Similarity in meaning can then be measured (e.g., using the cosine distance between word vectors) and expressed as a quantity.

The challenge now lies in deciding what degree of similarity is sufficient to the notion of synonymity, or in telling near-synonyms from "true" ones. Typically, there is no obvious cut-off that separates those sides consistently across many examples, and we need to trade off precision for recall. Another pitfall lies in the fact that antonyms (such as *big* and *small*) tend to occur in the same linguistic environments, such that word embedding algorithms often are unable to differentiate them and assign high similarity scores to them.

AMBIGUITY IN SENSE AND CATEGORY One of the greatest challenges in lexical simplification lies in the phenomenon that many words have more than one meaning (or 'sense'). Whether due to homonymy or (structural) polysemy, it is often difficult for computers, and sometimes even for humans, to confidently decide which of its senses a word bears in a given context. Being able to do so is an obvious necessity in lexical substitution. Considering the sentence *A bridge across the sound connects Denmark and Sweden*, we must not replace *sound* with *noise*, as it carries its 'body of water' meaning here. A more suitable substitution in this case would be *strait* or *sea passage*.

In some languages, including English, this is aggravated by the fact that many words can belong to different parts of speech. Again, *sound* is a good example, as it can be a noun, verb or adjective. The classic example, however, is *round*, which can be one of five parts of speech (a noun, a verb, an adjective, an adverb or a preposition). If we want to replace this word with a synonym, we need to know which part of speech it has in a specific context, since its synonyms are specific to only some of these categories (e.g, *lap* for the noun or *around* for the adverb). This phenomenon, called recategorization, is fairly productive in modern English and extends to other aspects of grammar, for instance noun subcategories such as abstract vs. concrete, mass vs. count, proper vs. common (Brinton and Brinton, 2010).

In the lexical simplification pipeline, this problem is typically tackled in the substitution selection step. Context awareness can for example be achieved by explicit word sense disambiguation. If we can link a specific occurrence of an ambiguous word to one of its senses as encoded in, for instance, WordNet, we can retrieve synonyms for this sense specifically. However, this comes with the typical problems of such a resource-reliant approach, including the limited coverage or unavailability of such resources in most languages. An alternative approach is suggested by Biran et al. (2011), who measure the cosine similarity between a synonym candidate and the target's context, and only allow the substitution if the similarity is high, assuming that its high semantic relatedness implies that the target sense is synonymous with the replacement. A simpler approach is the default strategy in substitution selection, namely the scoring of the altered phrase with a statistical language model.

MORPHOLOGICAL VARIATION Depending on the language a lexical simplification system works on, morphological variation of words can pose a major challenge. Keywords in synonym dictionaries are typically limited to lemmas, such that inflected forms have to be reduced to their lemmas before retrieving synonyms. Once this hurdle is overcome, the next challenge is to inflect the synonym lemma for the morphological features stripped in the first step. These two problems can be addressed with morphological analyzers, but the availability and quality of these is very limited across languages. For English, however, this approach has been explored by Lal and Ruger (2002), while Biran et al. (2011) generate a synonymy list that directly includes inflected forms and their inflected synonyms using Morphadorner (Burns, 2013).

Another problem is caused by syncretisms, i.e., identical surface forms that are grammatically ambiguous. As an example, consider the Danish word *tegn* ('sign', 'signal'), which in this form can either be singular or plural. In the Danish WordNet (Pedersen et al., 2009), *signal* ('signal') is listed as a synonym. If we now want to substitute *signal* for *tegn*, we need to decide whether to inflect the former in its singular form (*signal*) or in its plural form (*signaler*). A way to disambiguate between singular and plural could be through deep grammatical analysis or using a probabilistic language model, but neither of these methods is always reliable.

grammaticality Besides the issue of morphological variation outlined above, grammaticality can be corrupted when two synonymous words display different syntactic behavior, for instance when verbs take different semantic roles or through differences in selectional preferences. A related problem occurs when phrasal verbs are not recognized as such during CWI. For instance, in example 5a, failing to detect *cares for* as a multi-word unit and only replacing *cares* may produce ungrammatical (5b) output or distort meaning (5c). In order to ensure a valid paraphrasation as in example 5d, we need to identify the entire phrase as the replacement target.

(5) a. She **cares for** elderly patients.

b. * She minds for elderly patients.

c. She looks for elderly patients.

d. She looks after elderly patients.

2.2 higher-level simplification

Over the last couple of decades, text simplification has received a significant amount of attention from the research community also beyond the lexical aspect. Adding to the early works that established the field and which have been briefly discussed above, this section focuses on work that tackles text simplification with data-driven approaches. Notably, all works discussed here operate on the sentence level as their basic input unit. In this way, simplification can be tackled in a very similar fashion to machine translation, which also typically works on sentences in isolation. In this section, we first give an account of focused efforts that aim to catch specific linguistic phenomena and simplify them, before turning to translation-based simplification strategies that are not engineered with specific phenomena in mind. Lastly, we discuss the still open challenges in sentence simplification research.

2.2.1 *Targeted simplification*

Text simplification strategies that focus on specific linguistic phenomena marked the first approaches to simplification in general. Here we give an account of the most important simplification targets that past research has dealt with.

compression and summarization One of the most explored ways to automatically reduce reading difficulty of sentences is to shorten them by removing words and phrases such that only the essential information is retained. The earliest work in this area is presented by Knight and Marcu (2000), who address the problem by scoring subtrees of the input sentence parse with a noisy-channel model. The model is induced from the Ziff-Davis corpus, a collection of sentences and manual compressions from newspaper articles about technological products.

Related to this approach is the one by Cohn and Lapata (2009), who also operate on parse trees. However, their method differs from the former by employing the Synchronous Tree Substitution Grammar framework, which Siddharthan (2014) points out would in principle allow to cover reorderings and insertions of subtrees, although this possibility is not explored. Grammar-based solutions to sentence compression are further proposed by Woodsend and Lapata (2011), Siddharthan and Mandya (2014), and Mandya et al. (2014).

An obvious problem with the reliance on parsers is their limited accuracy and the emergence of downstream errors from incorrect parsing output. Work that partly overcomes the necessity of high-quality parses was introduced by McDonald (2006), who suggests discriminative learning algorithms to score and decode from the full set of possible compressions. While this approach still uses syntactic information provided by parsers as features, parse errors can be assumed to have less dramatic effects on the final prediction. Indeed, this approach appears to produce more grammatical output than the previous works (McDonald, 2006).

Nomoto (2007) presents an even simpler approach that uses Conditional Random Fields to only operate on the flat input sequence and delete or retain individual tokens. This idea was later pursued by Filippova et al. (2015), who make use of the availability of significantly more training data which they mine from pairs of first sentences in newspaper articles and the respective headlines. Their model is a recurrent neural network (RNN) and poses the first approach to compression that makes use of deep learning strategies. Another notable contribution comes from Klerke et al. (2016), who extend the RNN-based model and use gaze signals from an external corpus to inform sentence compression in a multi-task learning setup.

Another line of work has tackled abstractive sentence summarization, which is the condensation of information that goes beyond merely deleting words and phrases, and employs rewriting strategies to generate an *abstract* of an input sentence. A recent and notable contribution is the work by Chopra et al. (2016), who use a recurrent neural network trained on sentence-headline pairs from the Gigaword corpus. Their encoder-decoder architecture is inspired by neural machine translation models, and in this respect is very similar to some of the work discussed in the next section, but their focus on sentence summarization sets it apart from most of that work. Further, note that like extractive sentence compression, abstractive summarization is often not primarily understood as a simplification strategy, but rather seen by many authors as an "important step towards natural language understanding" (Chopra et al., 2016).

PRONOUN REPLACEMENT Another simplification strategy that has received significant attention is the substitution of pronouns with their referents. Recovering the antecedent poses difficulty for a number of target populations, most notably people on the autism spectrum. For them, it is particularly challenging to interpret the cue given by the pronoun that a referent "was recently discussed and is available in memory, but is not currently in attention" due to their difficulty in directing and managing attention (O'Connor and Klein, 2004). This task is generally challenging for readers who suffer from a reduced processing bandwidth, for example people with dyslexia who need to dedicate a great deal of their attention to decoding letters to sounds.

These observations motivated the first works that explicitly address pronoun replacement as part of simplification systems. These were part of the PSET project (Carroll et al., 1998) and explicitly addressed people with aphasia (Canning and Tait, 1999) and dyslexia (Canning et al., 2000). Further work on pronoun replacement comes from Siddharthan (2003), who incorporates a model of a reader's attention to detect pronouns that are difficult to resolve. Finally, Yaneva and Evans (2015) lends further support for the effectiveness of replacing pronouns in text simplification systems for the benefit of people with autism.

SENTENCE SPLITTING Difficulty in reading comprehension is aggravated as sentences become longer and take a greater toll on the working memory of the reader. A very obvious way to reduce text complexity is therefore to split long sentences into several shorter ones, a strategy that is often encountered as one of the most frequent simplification operations in corpora of manually produced simplifications across a number of languages (Petersen and Ostendorf, 2007; Gasperin et al., 2009; Bott et al., 2012b; Xu et al., 2015). Sentence splitting has therefore been explored since the early works of Chandrasekar et al. (1996), Chandrasekar and Srinivas (1997), and Dras (1999). These approaches make use of pattern-based rewriting rules over parse trees, either hand-coded as in the first of these works or induced from manually produced simplifications in the case of the latter two.

A popular target for these early works were non-restrictive relative clauses, which the authors aimed to extract and then append, with the referents as their subjects, to the main sentence. Such pattern-based approaches manage to capture relative clauses reasonably well, but tend to exhibit problems in inferring the correct referent. Siddharthan (2003, 2006) points out these issues and employs models of discourse structure, most notably Centering theory (Grosz et al., 1995), to preserve text cohesion. This work also approaches sentence splitting beyond the extraction of relative clauses, covering apposition, coordination and subordination.

Zhu et al. (2010) follow earlier approaches in relying on parses and rules to identify possible splits, but they incorporate a statistical component into their method that learns if a split at a certain subordinate clause is valid. This decision is followed by a completion step, in which the split-off clause is transformed into a full sentence, e.g., by copying the subject from the main clause. In a similar vein, Lee and Don (2017) score splits at various locations in a parse tree by employing decision trees, yet their work neglects the necessary second step of completing the split-off material to a full sentence.

Finally, Bott et al. (2012b) present work on sentence splitting in Spanish that is, analogously to the English work cited above, based on syntactic rules.

OTHER EXAMPLES The simplification strategies listed above are only the most prominent ones. There is, however, a wide range of other types of sentence simplification that have been tackled by a smaller number of papers and research projects. These include the transformation of periphrastic *of*-possessives to genitive forms or the resolution of cleft constructions, both tackled by Dras (1999).

Another interesting approach is to augment the original sentence with additional material that explains difficult vocabulary and unfamiliar concepts such as technical terminology. Support for this strategy comes from Rello et al. (2013b), who show that in some cases this is preferable to substitution as it is normally done in lexical simplification.

2.2.2 *Simplification as monolingual translation*

An approach that has for some time been a major trend in simplification research is to treat the task as a special case of machine translation (MT) with identical source and target languages. A first contribution in this direction is the work of Zhu et al. (2010), which uses a translation model based on parse tree transformations, making use of the reduced necessity to perform translation operations at the lexical level in the case of monolingual translation. Subsequent developments in the simplification-as-translation area then focused on the statistical machine translation paradigm and later moved on to neural methods, reflecting general trends in MT. We very briefly discuss these different methods below and highlight a number of their applications to simplification, first for statistical and then for neural machine translation.

STATISTICAL MT In statistical MT, in particular the phrase-based kind (PBMT), a system first learns alignments between words and multi-word units (phrases) from a corpus of parallel sentences. From the same data, translation likelihoods between these phrases are then estimated. At decoding time, a system generates partial translations from an input and scores them based on its translation model as well as a language model that is supposed to ensure fluency and grammaticality in the output. For more detail, see Koehn (2009).

Approaches that have followed this paradigm for the purpose of text simplification have in some cases used a standard PBMT setting (Specia, 2010) or adapted the decoder to accommodate certain desiderata in simplification, for instance promoting diversity between the input and output (Wubben et al., 2012) or keeping the output short (Coster and Kauchak, 2011b).

A challenge in translation-based simplification has been the scarcity of parallel data. The previously mentioned strategy of extracting pairs of standard and simplified sentences from the Simple English Wikipedia provided data for many of the first translation-based approaches in text simplification for English. Yet, besides its lack of clear guidelines and professional editors, the drawbacks of this dataset quickly become apparent when we investigate more closely the output produced by many of these phrase-based approaches. As the findings by Alva-Manchego et al. (2017) suggest, the generally high similarity between the input and output sentences — in particular on the more local level of words and phrases, since in the majority of cases these are retained — makes learning abstract simplification operations difficult for a standard PBMT system that makes many local decisions.

In response to such problems, Xu et al. (2016) propose ways to adapt statistical machine translation systems to the simplification task. Their contributions include the development of simplification-specific objective functions and metrics as well as a method to score paraphrase rules for simplicity.

NEURAL METHODS The rise of neural methods in machine translation, propounded by the seminal works of Kalchbrenner and Blunsom (2013), Sutskever et al. (2014), and Cho et al. (2014), has led a number of researchers to try to adopt this technology for text simplification. Neural MT systems usually consist of two main components, an encoder and a decoder. The former iterates over a sequence of words that are represented through high-dimensional embedding vectors and computes a state vector across every timestep, such that at the end of the iteration the state vector is a fixed-size representation of the entire sequence (typically a sentence). The decoder then generates output words based on this input representation as well as the previously generated output. Later developments have highlighted the benefit of an attention mechanism that lets the decoder focus on specific parts of the input sentence (Bahdanau et al., 2014; Luong et al., 2015). The advantages of the neural approach over statistical MT lie in more fluent and grammatical output as well as better capturing of long-range dependencies, albeit at the cost of slower model training and the need for large volumes of training data due to the large number of parameters typically involved.

Through the availablity of the Newsela corpus (Xu et al., 2015), neural MT has become a feasible alternative in sentence simplification.[5] Nevertheless, the first work that used neural MT methods for text simplification Nisioi et al. (2017b) still chooses to learn simplifications from the Simple Wikipedia corpus compiled by Hwang et al. (2015). Zhang and Lapata (2017) use the Newsela data and extend the former work by optimizing directly for simplification-relevant metrics, most notably SARI (Xu et al., 2016), through reinforcement learning. Finally, Scarton and Specia (2018) propose a neural simplification model which takes as an additional input a desired simplicity level, which to some degree allows them to adapt the output to specific readers.

2.2.3 *Challenges in sentence simplification*

OUTPUT GRAMMATICALITY A major risk in manipulating text is that it can be rendered ungrammatical. This applies to all simplification approaches, whether a text simplification system exchanges a word with a suspected synonym, removes words or entire phrases from a sentence, or changes its grammatical structure. The potential causes of ungrammatical output are manifold and can range from preprocessing errors (e.g., incorrect parses in a heavily syntax-reliant system) to naive substitutions of phrases that would require different syntactic configurations (see also the preceding discussion on challenges in lexical simplification).

Ungrammaticality affects text simplification more than other text generating applications, for instance chatbots or translation systems. In comparison to those, ungrammatical output has much more severe repercussions in text simplification, where the user is generally more sensitive to surprising and uncanonical language. Grammatical errors and semantic anomalies eventually make text less intelligible (or perhaps even completely unintelligible), thereby causing a system's

[5] The corpus is aligned at the document level, such that parallel sentences need to be extracted first, and the parallel corpus size ultimately depends on this process. However, the available number of parallel sentences is typically much larger than Wikipedia-based datasets, though much smaller than standard MT corpora.

output to be entirely diametrical to the principal and ultimate goal of text simplification, namely to promote accessibility.[6]

Ensuring grammaticality is not very straightforward, and the detection of grammatical errors is an active research field in its own right (Chodorow and Leacock, 2000; Gamon, 2011; Leacock and Chodorow, 2003; Cummins and Rei, 2018). Papers on text simplification often discuss grammaticality issues and evaluate for grammaticality, but in general no solutions to this problem are provided beyond the application of statistical language models to re-rank hypotheses for fluency. This approach, however, is generally not good enough, since it only provides relative comparisons between hypotheses and cannot reliably judge grammaticality. Further research in simplification is thus advised to work towards this goal of ensuring grammatical output.

RESOURCE SCARCITY The limited availability of resources for text simplification has been a problem for the field, even if individual contributions such as the development of a simplification-specific evaluation metric (Xu et al., 2016) or the publication of a more targeted simplification corpus (Xu et al., 2015) have had a notable impact and have partly solved some of the issues pointed out by Shardlow (2014a). However, most simplification datasets are still limited to English and are still relatively small in comparison to datasets in other areas of NLP, such that many of the technical advances used in technically related areas like machine translation cannot be easily transferred to simplification.

The challenge for text simplification research lies in developing solutions, in particular in low-resourced languages, to overcome the need for great amounts of data. Approaches that could work in this direction include transfer and multi-task learning across tasks and languages, see also chapters 7 and 8 of this dissertation. Further, the work presented in chapters 5 and 6 investigates methods to continuously receive explicit and implicit input from users, such that existing datasets can be augmented with this information.

[6] Note, however, that in special circumstances, ungrammatical sentences are more readily acceptable than grammatical ones, with certain grammatical phenomena giving rise to a *perceived grammaticality* (Gibson and Thomas, 1999; Shravan et al., 2010)

Part II

ADAPTABLE MODELS FOR SENTENCE SIMPLIFICATION

3 TEXT SIMPLIFICATION AS TREE LABELING

ABSTRACT

We present a new, structured approach to text simplification using conditional random fields over top-down traversals of dependency graphs that jointly predicts possible compressions and paraphrases. Our model reaches readability scores comparable to word-based compression approaches across a range of metrics and human judgements while maintaining more of the important information.

3.1 INTRODUCTION

Sentence-level text simplification is the problem of automatically modifying sentences so that they become easier to read, while maintaining most of the relevant information in them. This can benefit applications as pre-processing for machine translation (Bernth, 1998) and assisting technologies for readers with reduced literacy (Carroll et al., 1999; Watanabe et al., 2009; Rello et al., 2013a).

Sentence-level text simplification ignores sentence splitting and reordering, and typically focuses on *compression* (deletion of words) and *paraphrasing* or *lexical substitution* (Cohn and Lapata, 2008). We include paraphrasing and lexical substitution here, while previous work in sentence simplification has often focused exclusively on deletion. Approaches that address compression and paraphrasing (or more tasks) integrally include (Zhu et al., 2010; Narayan and Gardent, 2014; Mandya et al., 2014).

Simplification beyond deletion is motivated by the observation in Pitler (2010) that abstractive sentence summaries written by humans often "include paraphrases or synonyms ('said' versus 'stated') and use alternative syntactic constructions ('gave John the book' versus 'gave the book to John')." Such lexical or syntactic alternations may contribute strongly to the readability of a sentence if they replace difficult words with shorter or more familiar ones, in particular for low-literacy readers (Rello et al., 2013a). Our joint approach to deletion and paraphrasing works against the limitation that abstractive simplifications "are not capable of being generated by [...] most sentence compression algorithms" (Pitler, 2010).

Furthermore, a central concern in text simplification is to ensure the grammaticality of the output, especially with low-proficiency readers as the target audience. Our approach to this problem is to remove or paraphrase entire syntactic units in the original sentence, thus avoid-

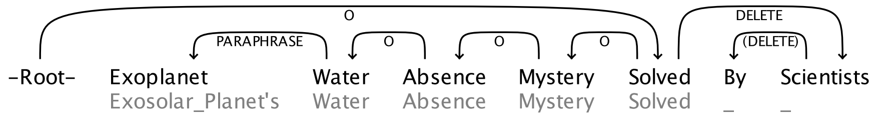


Figure 3.1: An example simplification tree

ing to remove phrase heads without removing their arguments or modifiers. Like Filippova and Strube (2008), we rely on dependency structures rather than constituent structures, which promises more robust syntactic analysis and allows us to operate on discontinuous syntactic units.

CONTRIBUTIONS We present a sentence simplification model which is, to the best of our knowledge, the first model that uses structured prediction over dependency trees *and* models compression and paraphrasing jointly. Our model uses Viterbi decoding rather than scoring of all candidates and outputs probabilities reflecting model confidence.

3.2 DATA

We use the publicly available Google compression data set,[7] which consists of 10,000 English sentence triples with (1) the original sentence as present in the body of an online news article, (2) a headline based on the original sentence, and (3) a compression that is automatically derived from the original such that it only contains word forms present in the original, preserving their order. The following sentence triple exemplifies these different versions:

(1) *In official documents released earlier this month it appears the Queen of England used the wrong name for the Republic of Ireland when writing to president Patrick Hillery.*

(2) *Queen elizabeth ii used wrong name for Republic*

(3) *The Queen of England used the wrong name for the Republic of Ireland.*

The data is pre-processed with the Stanford CoreNLP tools (Manning et al., 2014), retrieving lemmas, parts-of-speech, named entities and dependency trees. We reserve the first 200 sentences from the data set for evaluation, the next 200 for tuning parameters (including the used PPDB versions, see next paragraph), and use the remaining 9,600 sentences for training our model.

[7] http://storage.googleapis.com/~sentencecomp/compressiondata.json

DELETION AND PARAPHRASE TARGETS As our approach operates on dependency trees, aiming to prune or paraphrase subtrees from the dependency tree of a sentence, we identify deleted or paraphrased subtrees, marking their heads with a corresponding label. A subtree receives a Delete label if none of the words subsumed by this subtree occur in the compressed version of the sentence.

We identify paraphrased subsequences in an original sentence by looking up the subsequence string in the Paraphrase Database (PPDB) (Ganitkevitch et al., 2013) and testing if one of its possible paraphrases occurs in the headline version of the sentence in question. The Paraphrase Database 1.0 is a set of phrasal and lexical pairs that were automatically acquired from bilingual parallel corpora, and thus contain a portion of flawed paraphrase pairs. The database comes in a number of different sizes, where small editions are restricted to high-precision paraphrases with relatively high paraphrase probabilities. As the two smallest editions of PPDB only yield a very low number of paraphrase targets (less than 100 in the entire Google compression data set), we opt to employ a medium-sized version of the resource (size 'L') and find a total of 510 phrasal and lexical paraphrases in the corpus.

3.3 METHOD

We assume that text simplification is a generative process on syntactic dependency graphs with a paraphrase dictionary. A dependency graph $G = (V, A)$ is a labeled directed graph in the standard graph-theoretic sense and consists of nodes, V, and arcs, A, such that for sentence $S = w_0 w_1 \ldots w_n$ and label set R, $V \subseteq \{w_0, w_1, \ldots, w_n\}$, and $A \subseteq V \times R \times V$ hold, and if $(w_i, r, w_j) \in A$ then $(w_i, r', w_j) \neq A$ for all $r' \neq r$. We restrict the dependency graphs to the class of trees, i.e., for $(w_i, r, w_j) \in A$, if $(w_k, r, w_j) \in A$ then $k = i$.

The generative process traverses the tree in a top-down fashion, deleting or paraphrasing subtrees (see Figure 3.1). Note that elements in subtrees dominated by a deleted node are automatically deleted (analogously for paraphrases).

For each dependency tree $G = (V, A)$ in a training set of T sentences, we derive an input sequence of K-dimensional feature vectors $\mathbf{x} = x_1, \ldots, x_n$ and an output sequence of $\mathbf{y} = y_1, \ldots, y_n$. Our tree-to-string simplification model is a second-order linear-chain conditional random field (CRF)

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{i=1}^{n} \exp\{\sum_{k=1}^{K} \theta_k f_k(y_t, y_{t-1}, x_t)\}$$

with $y_i = \text{Delete}$ if and only if x_i represents the least upper bound in G covering a deleted span in the training data, and $y_i = \text{Paraphrase}$ if and only if x_i represents the least upper bound in G covering a paraphrased span in the training data. For example, if the entire sentence is deleted, and $(w_0, r, w_i) \in A$, then $y_i = \mathsf{Delete}$ (but $y_j = \mathsf{Leave}$ for $j \neq i$).

This encoding means that theoretically we can predict to paraphrase a subtree that is dominated by a node which is in turn predicted to be deleted (or vice versa). However, once an operation is carried out on a subtree, none of its dominated nodes are considered in the remainder of the top-down simplification process. Giving preference to operations at higher-level syntactic environments in this manner serves as a mechanism to resolve ambiguities in the decision process by taking a wider context into account.

Furthermore, predicting a node to get paraphrased at the right corner of a deleted subtree can potentially influence labeling decisions outside this subtree as a consequence of the dynamic-program Viterbi decoding. We acknowledge that this is a theoretical drawback of the presented approach, but given that we do not observe any such dependency graphs in our data, we do not expect this to be a serious problem in most cases.

Whenever our model predicts that a subtree be paraphrased, we look up the respective token sequence in PPDB and replace it with the candidate paraphrase (if available) that maximises the product of frequency and translation probability according to PPDB.

FEATURES FOR CRF MODEL We train a second-order CRF model using MarMoT (Mueller et al., 2013), an efficient higher-order CRF implementation. The model computes its observational probabilities from features based on properties of the subtree root token (incl. POS, language model probability, NE mention, word difficulty), of the internal structure of the subtree (incl. number of children, depth, length of sequence), and of the external grammatical structure (incl. dependency relation, parent POS, distance from parent, position in sentence).

3.4 EVALUATION

BASELINES In the following experiments, we compare our work to state-of-the-art approaches to sentence compression and joint compression/paraphrasing. For the first of these two categories, we consider the LSTM system described in Filippova et al. (2015) as well as the results reported therein for the MIRA system (McDonald, 2006). As a joint approach, we consider Reluctant Trimmer (RT), a simplification system that employs synchronous dependency grammars (Mandya et al., 2014). Since the LSTM system requires great amounts of training data, which were not available to us, we cannot reproduce

		Recall	**Precision**	**F1**
	Reluctant Trimmer			
tokens	Delete	54.60	20.23	29.52
	Paraphrase	01.67	66.67	03.27
	Leave	52.27	78.54	54.60
	Tree Labeling			
subtrees	Delete	43.31	67.54	52.77
	Paraphrase	23.85	50.89	32.48
	Leave	94.29	84.82	89.30
tokens	Delete	49.67	77.16	60.44
	Paraphrase	21.16	51.52	30.00
	Leave	80.33	50.91	62.32

Table 3.1: Performance on joint deletion and paraphrasing detection for our tree labeling system (evaluating both on entire subtrees and token level) as well as for the RT baseline (tokens only). Note that RT is trained on the (Simple) English Wikipedia, not on the Google compressions, and therefore the results may not be directly comparable.

its output and therefore limit our comparison of human rankings to the eleven output examples provided in the paper.

F-SCORES We first evaluate our tree labeling model (TL) on its ability to predict subtree deletion and paraphrasing (i.e. whether a subtree should be paraphrased, independent of the actual replacement). The results for this evaluation setup, as well as word-level performance, are listed in Table 3.1 and compared to RT. Note that for deletion and paraphrasing, our model consistently has higher precision than recall, thus generating more confident simplifications and less ungrammatical output.

AUTOMATED READABILITY SCORES Table 3.2 reports the compression ratio (CR, percentage of retained words) as well as automated readability scores that our model achieves on the test set and compares it to the output of the RT baseline. Our system manages to compress the original texts by more than one third, but the gold simplifications (headlines and compressions) are still considerably shorter.

Data version	CR↓	Flesch↑	Dale-C.↓
Original	—	49.15	9.55
Headlines	**0.32***	-80.77*	17.61*
Compressions	0.40*	**70.80***	9.56
TL output	0.62*	56.25*	9.30*
RT output	0.86*	60.65*	**9.27***

Table 3.2: Compression ratios and automatic readability scores for the Google compression data set, compared to the system output. Readability is indicated by a high Flesh Reading Ease score and a low Dale-Chall score. * indicates differences compared to the original sentences that are significant at $p < 10^{-3}$.

System	Readability	Informativeness
MIRA	4.31	3.55
LSTM	**4.51**	3.78
TL	4.14	**4.01**
RT (11)	3.09	4.12
LSTM (11)	**4.23**	3.42
TL (11)	4.21	**4.15**

Table 3.3: Mean readability and informativeness ratings for the first 200 sentences in the Google data (upper) and for the 11 sample sentences listed in Filippova et al. (2015) (lower).

Our approach improves readability as measured by the Flesch Reading Ease score[8] (Flesch, 1948) and the Dale-Chall formula (Dale and Chall, 1948). The former score measures textual difficulty as a function of sentence length and the number of syllables per word, while the latter aims to estimate a US school grade level at which a text can be well understood, based on a vocabulary list. Both metrics deem the output of our system easier to read than the original texts, while the Dale-Chall formula also rates our system better than the gold simplifications.

HUMAN READABILITY RATINGS Following Filippova et al. (2015) in their evaluation setup for the sake of comparability, we ask raters to assign scores on a one-to-five Likert scale to the first 200 sentences from the Google compression data paired with the output of our sys-

[8] The negative value that the headlines receive for this metric is due to an over-representation of longer words in headlines.

tem. Each pair is rated by three native or near-native speakers of English.

The raters are asked to evaluate the sentence pairs for *readability* and *informativeness*. The former, following Filippova et al. (2015), "covers the grammatical correctness, comprehensibility and fluency of the output." The latter metric pertains to the relation between the original sentence and the system output as it "measures the amount of important content preserved in the compression."

Table 3.3 compares the performance of our model to the figures reported in Filippova et al. (2015) for their LSTM model and McDonald's McDonald (2006) system (MIRA). For a comparison with the same judges, we repeat the evaluation with the 11 sample output compressions listed in Filippova et al. (2015) as well as the respective output from Reluctant Trimmer; see the lower part of Table 3.3. The results suggest that, compared to the compression-only LSTMs, our approach yields comparable performance in terms of readability, while maintaining more of the central information in the original sentences. Compared to RT, our system does considerably better in terms of readability and retains slightly more of the important information.

3.5 RELATED WORK

Several approaches to sentence compression have been presented in the last decade. Knight and Marcu (2002) and Turner and Charniak (2005) apply noisy channel models, using language models to control for grammaticality. McDonald (2006) introduces a different approach, discriminatively training a scoring function, informed by syntactic features, to score all possible subtrees of a sentence. His work was inspired by Riezler et al. (2003) scoring substrings generated from LFG parses. A third approach to sentence compression is sequence labeling, which has been explored by Elming et al. (2013) using linear-chain CRFs with syntactic features, and more recently by Filippova et al. (2015) and Klerke et al. (2016) using recurrent neural networks with LSTM cells.

Most recent approaches to sentence compression make use of syntactic analysis, either by operating directly on trees (Riezler et al., 2003; Nomoto, 2007; Filippova and Strube, 2008; Cohn and Lapata, 2008, 2009) or by incorporating syntactic information in their model (McDonald, 2006; Clarke and Lapata, 2008). Recently, however, Filippova et al. (2015) presented an approach to sentence compression using LSTMs with word embeddings, with no syntactic features. We return to working directly on trees, presenting a tree-to-string model of sentence simplification. Our model has interesting similarities to (Riezler et al., 2003), but uses Viterbi decoding rather than scoring of

	Original Sentence & *Simplifications*
O	OG&E is warning customers about a prepaid debit card scam that is targeting utility customers across the county.
C	*OG&E is warning customers about a scam.*
R	*OG&E is warning customers about a debit card scam that is targeting utility customers across the country.*
T	*OG&E is warning customers* ***regarding*** *a prepaid debit card scam.*
O	The husband of murdered Melbourne woman Jill Meagher will return to Ireland later this month "to clear his head" while fighting for parole board changes.
C	*The husband of murdered woman Jill Meagher will return to Ireland.*
R	*The husband of Melbourne woman Jill Meagher will return to Ireland this month to clear his head fighting for parole board changes.*
T	*The husband of murdered Melbourne woman Jill Meagher will return to Ireland.*
O	A research project has found that taxi drivers often don't know what the speed limit is.
C	*Taxi drivers don't know the speed limit is.*
R	*A research project has found that drivers often* ***do not*** *know what the speed limit is.*
T	*A project has found taxi drivers don't know what the speed limit is.*

Table 3.4: Example output for original sentences (O) as generated by the Reluctant Trimmer baseline (R) and our tree labeling system (T), as well as the headline-generated Google compressions (C).

all candidates. Also, it follows Cohn and Lapata (2008) in going beyond most of these models, modeling compression *and* paraphrasing.

For lexical simplification, most systems typically use pre-compiled dictionaries (Devlin, 1999; Inui et al., 2003) and select the synonym candidate with the highest frequency. More recently, Baeza-Yates et al. (2015) introduced an algorithm for lexical simplification in Spanish that selects the best synonym candidate in a context-sensitive fashion.

Cohn and Lapata (2008), Woodsend and Lapata (2011) and Mandya et al. (2014) present *joint* approaches to compression and paraphrasing that are based on (quasi-) synchronous grammars, and similarly Zhu et al. (2010) take a syntax-based approach, but employ a probabilistic model of various simplification operations. Napoles et al. (2011) do not use syntactic information, but instead employ a character-based metric to compress and paraphrase.

3.6 CONCLUSION

We presented a new approach to sentence simplification that uses linear-chain conditional random fields over dependency graphs to jointly predict compression and paraphrasing of entire syntactic units. The objective of our model is to delete or paraphrase entire subtrees in dependency graphs as a strategy to avoid ungrammatical output. Our approach makes innovative use of a three-fold parallel monolingual corpus that features headlines and compressions to learn paraphrases and deletions, respectively. Human evaluation shows that our approach leads to readability figures that are comparable to previous state-of-the-art approaches to the more basic sentence compression task, and better than previous work on joint compression and paraphrasing. While our model does rely on syntactic analysis, it only needs a tiny fraction (less than 0.5%) of the training data used by Filippova et al. (2015).

ACKNOWLEDGMENTS

This research was partially funded by the ERC Starting Grant LOWLANDS No. 313695, as well as by Trygfonden.

4

LEARNING HOW TO SIMPLIFY FROM EXPLICIT LABELING OF COMPLEX-SIMPLIFIED TEXT PAIRS

ABSTRACT

Current research in text simplification has been hampered by two central problems: (i) the small amount of high-quality parallel simplification data available, and (ii) the lack of explicit annotations of simplification operations, such as deletions or substitutions, on existing data. While the recently introduced Newsela corpus has alleviated the first problem, simplifications still need to be learned directly from parallel text using black-box, end-to-end approaches rather than from explicit annotations. These complex-simple parallel sentence pairs often differ to such a high degree that generalization becomes difficult. End-to-end models also make it hard to interpret what is actually learned from data. We propose a method that decomposes the task of TS into its sub-problems. We devise a way to automatically identify operations in a parallel corpus and introduce a sequence-labeling approach based on these annotations. Finally, we provide insights on the types of transformations that different approaches can model.

4.1 INTRODUCTION

Text Simplification (TS) is the task of reducing the complexity of a text without changing its meaning. Simplification can be applied at various linguistic levels, from lexical substitution to more global operations such as sentence splitting, paraphrasing or the deletion or reordering of entire clauses.

Existing corpora for TS generally come in one of two variants. The first focuses on very specific sub-problems, such as sentence compression (Bingel and Søgaard, 2016) or the identification of difficult words (Paetzold and Specia, 2016b), and typically encodes relevant simplification operations as discrete labels on tokens. The other variant includes more general, higher-level types of simplifications that often entail the rephrasing or re-structuring of sentences, with content added or removed. These “natural" simplifications are often created for end-users rather than for research purposes. Examples of the latter simplification resources include the Newsela (Xu et al., 2015) and Simple English Wikipedia corpora (Zhu et al., 2010; Coster and Kauchak, 2011c). These resources generally encode interdependencies between different types of simplification better than single-purpose resources and may thus seem favorable for learning simplifications.

However, the freedom given to editors and lack of explicit labels on the modifications performed makes generalization much more difficult, especially when existing resources are relatively small in comparison to corpora for other text-to-text problems like machine translation (MT). Nevertheless, these corpora have been extensively used to learn phrase-based statistical and neural models for end-to-end TS systems that bear resemblance to MT models (Specia, 2010; Zhu et al., 2010; Coster and Kauchak, 2011c; Wubben et al., 2012; Narayan and Gardent, 2014; Xu et al., 2016; Zhang and Lapata, 2017; Zhang et al., 2017; Nisioi et al., 2017a).

ADAPTABILITY AND INTERPRETABILITY MT-style models are essentially black boxes that offer little or no control over the way in which a given input is modified. Additionally, in most cases the types of modifications that are actually learned are limited to paraphrasing of short sequences of words. We believe a middle ground is missing in terms of resources and approaches for TS, where models are learned from a more informed labeled dataset of natural simplifications, and can then be applied in a controlled way, e.g., in adaptive simplification scenarios that prioritize different ways of simplifying (e.g. compression or sentence splitting) depending on a particular user's needs.

The only previous work on TS via explicitly predicting simplification operations is that by Bingel and Søgaard (2016), who create training data from comparable text to label entire syntactic units and train a sequence labeling model to predict deletions and phrase substitutions in a complex sentence. Our approach is different in that it captures a larger variety of operations in a more global fashion, by using sentence-wide word alignments rather than surface heuristics. Furthermore, we use a more reliable (professionally created) corpus and our approach is more flexible as we do not rely on syntactic parse trees at test time.

CONTRIBUTIONS This paper introduces the following main contributions: **(1)** We provide an in-depth analysis on the potential and limitations of the dominant approach to TS: end-to-end MT-style models; **(2)** We devise a method to automatically identify specific simplification operations in aligned sentences from complex-to-simple simplification corpora. This results in a corpus that can be used to study how human experts perform simplification tasks, as well as to train simplification models to address specific problems; and **(3)** We propose a sequence labeling model built from such a corpus to predict which simplification operations should be performed as a first step for a complete simplification pipeline. This approach is highly modular: once operations are identified, different methods can be applied to cover each simplification operation. We show that this operation-based TS approach is able to produce simpler texts than end-to-end models. The code for extracting the simplification operations is available at https://github.com/ghpaetzold/massalign, while our sequence labeling model is released at https://github.com/jbingel/ijcnlp2017_simplification.

4.2 RELATED WORK

In what follows we give a brief description of previous work on statistical and neural models for TS. We first compare methods using versions of Simple English Wikipedia data (Zhu et al., 2010; Coster and Kauchak, 2011c), before considering recent work that relies on the professionally edited Newsela corpus (Xu et al., 2015).

SIMPLE ENGLISH WIKIPEDIA Zhu et al. (2010) propose a syntax-based translation model for TS that learns operations over the parse trees of the complex sentences. They outperform several baselines in terms of Flesch index. Coster and Kauchak (2011c) train a phrase-based machine translation (PBMT) system and obtain significant improvements in terms of BLEU (Papineni et al., 2002) over a baseline. Coster and Kauchak (2011a) extend a PBMT model to include phrase deletion and outperform Coster and Kauchak (2011c). Wubben et al. (2012) also train a PBMT system for TS with a dissimilarity-based re-ranking heuristic, outperforming Zhu et al. (2010) in terms of BLEU. Narayan and Gardent (2014) built TS systems by combining discourse representation structures with a PBMT model, which outperforms previous approaches. Xu et al. (2016) modify a syntax-based MT system in order to use a new metric – SARI – for optimization and to include special rules for paraphrasing. Although their system does not outperform previous work in terms of BLEU, it achieves the best results according to SARI and human evaluation. Zhang et al. (2017) train a lexically constrained sequence-to-sequence neural network model for TS, based on the encoder-decoder architecture for MT. The system outperforms baseline systems (including a PBMT system) in terms of BLEU. Finally, Nisioi et al. (2017a) propose a model for TS that is able to perform lexical replacements and content reduction. They use a neural encoder-decoder approach where they combine pre-trained (general domain and in-domain) word embeddings for the source and target sentences. They also perform beam search, finding the best beam size using either BLEU or SARI. Their best model outperforms previous PBMT-based approaches in terms of BLEU.

NEWSELA CORPUS To the best of our knowledge, Zhang and Lapata (2017) is the only work that explores MT-based approaches on the Newsela corpus. They train an attention-based encoder-decoder model (Bahdanau et al., 2014) and use reinforcement learning with a reward policy combining SARI, BLEU and cosine similarity (to measure meaning preservation). Their approach shows improvements over a PBMT system in terms of BLEU and SARI, but no insights are given with respect to the transformations that are actually learned or how distant from the original sentences the simplifications are. They also experiment with the Simple Wikipedia corpus, yet do not outperform Narayan and Gardent (2014) on this data.

The neural end-to-end model we implement as a baseline in this paper is equivalent to that in Zhang et al. (2017) without the lexical constraints, while the statistical model is equivalent to the one in Coster and Kauchak (2011c).

4.3 SIMPLIFICATION VIA END-TO-END MODELS

In addition to requiring large amounts of training data, MT-based approaches to TS are limited because of their black-box way of addressing the problem. As we are going to show in this section, standard end-to-end systems without special adaptation to TS do not succeed in learning alternative formulations of the original text. With a few exceptions (by the neural model), they tend to repeat the original text. We conjecture that this is because, for most original-side material, TS corpora do not consistently enough offer alternative simplified formulations: in the majority of instances, most words are kept as in the original.

To study the potential and limitations of end-to-end translation models for TS, we build models using state-of-the-art MT-based approaches and the Newsela corpus, arguably the most reliable (professionally created) and realistic (aimed at a target audience rather than research) resource to date.

THE NEWSELA CORPUS Newsela is a multi-comparable corpus, where each document comes in up to six levels of simplicity, from 0 (original) to 5 (simplest).[9] In our experiments, we only use sentence pairs stemming from adjacent levels of simplicity within the same document.[10]

Translation approaches require data aligned at the sentence level. Given the original Newsela corpus, which only aligns different versions of the same document, we first align sentences using the algo-

[9] The Newsela Article Corpus was downloaded from `https://newsela.com/data`, version 2016-01-29.

[10] The motivations for only using adjacent levels are (i) that we assume that these are not "naturally" created (i.e. an expert would not start from an original text and directly generate a level 5 text, but rather go from 0 to 1, 1 to 2, ..., 4 to 5), and (ii) that the high degree of linguistic and stylistic differences between non-adjacent levels makes learning even more complex. For example, the average edit distance for sentences in the 0-1 group is 0.19, while for sentences in the 0-5 group, it is 0.65. As far as the first reason is concerned, note that we could not find any publicly available simplification guidelines for the Newsela corpus.

rithms described in (Paetzold and Specia, 2016e). Their algorithms search for the best alignment path between the paragraphs and sentences of parallel documents based on TF-IDF cosine similarity and an incremental vicinity search range. They address limitations of previous strategies (Barzilay and Elhadad, 2003; Coster and Kauchak, 2011c; Smith et al., 2010; Xu et al., 2015; Bott and Saggion, 2011) by disregarding the need for (semi-) supervised training, allowing long-distance alignment skips, and capturing N-to-N alignments. The alignments produced are categorized as:

- **Identical:** The alignment is one-to-one and the sentences are exactly the same (96,909 pairs across all adjacent levels).
- **1-to-1:** The alignment is one-to-one and the original-simplified sentences are different (130,790 pairs across all adjacent levels).
- **Split:** The alignment is 1-to-N (42,545 pairs across all adjacent levels).
- **Join:** The alignment is N-to-1 (7,962 pairs across all adjacent levels).

TRANSLATION MODELS. We built two types of models using state-of-the-art MT-based approaches: a phrase-based statistical MT model using Moses (Koehn et al., 2007),[11] and a neural MT model using Nematus (Sennrich et al., 2017).[12] The Neural Text Simplification tool (NTS) made available by Nisioi et al. (2017a) was also used for comparison.[13]

For our translation-based experiments, we consider two combinations of sentence alignments, using (i) only one-to-one alignments (**1-to-1**) (130,970 sentence pairs), and (ii) all alignments (**all**), i.e., the entire sentence-aligned corpus with identical, 1-to-1, split and join alignments (278,206 sentence pairs). The first type of data (1-to-1) is the focus of this paper (see §4.4). The latter variant is included in the experiments for comparison, in particular to address the question whether more (but not necessarily better) data can aid data-intensive translation-based approaches. For all experiments, the respectively used data was first randomly split into training (80%), development (10%) and test (10%) sets and normalized for entities (incl. names, locations, numbers).

SIMPLIFICATION QUALITY. The first and second sections of Table 4.1 show the results of translation-based systems according to several metrics: similarity metrics commonly used in MT, comprising BLEU (Papineni et al., 2002) and TER (Snover et al., 2006, minimum edit distance), as well a specific text simplification metric, SARI (Xu

[11] We follow instructions from `http://www.statmt.org/moses/?n=Moses.Baseline`

[12] We use a vocabulary size of 30,000 and the same parameters as in Sennrich et al. (2016).

[13] We use the same configurations as Nisioi et al. (2017a).

	Hyp vs. Ref		Hyp vs. Orig			
System	BLEU↑	TER↓	BLEU	TER	%Same↓	SARI↑
Moses (all)	69.64	30.20	98.77	0.41	93.03	27.45
Nematus (all)	36.46	52.66	45.40	42.30	21.60	22.91
NTS (all)	68.35	31.37	90.52	7.19	72.91	27.36
Moses (1-to-1)	57.79	40.19	98.30	0.86	89.50	24.58
Nematus (1-to-1)	46.90	52.84	76.29	20.10	30.45	29.89
NTS (1-to-1)	53.79	45.24	77.63	16.70	42.76	30.44
Silver op. (1-to-1)	67.33	22.66	61.63	26.01	10.83	61.71
Predicted op. (1-to-1)	41.37	48.72	59.71	25.24	14.06	31.29

Table 4.1: Performance of translation-based and operation-based TS models (using silver or predicted operation labels, with only DELETION and REPLACE applied). Metrics are BLEU and TER between simplified version (Hyp) and reference (Ref) or original version (Orig), the percentage of sentences copied from the input (%Same), and SARI for the simplifications.

et al., 2016). SARI measures how good the words added, deleted and kept by a simplification system are, after comparing the produced output to the original sentence and the simplification reference(s). It is similar to BLEU but rewards copying words from the original sentence. According to experiments performed by Xu et al. (2016), SARI is the metric that best correlates with human judgments of simplicity.

For both "all" and "1-to-1" variants, the BLEU and TER scores between hypotheses and references are worse for Nematus, showing that a baseline neural model tends to be more aggressive and potentially generate noisier modifications than Moses equivalents. To measure how strongly the various approaches modify the input sentences, these scores are also reported between the generated simplifications and the original inputs. Again, these metrics are worse for Nematus-based models, showing that they indeed perform more modifications on the sentences. Moses in turn is very conservative, keeping 90-93% of the test sentences exactly in their original version. SARI shows low scores for all systems. NTS is also conservative in the "all" variant (attested by the high BLEU score between hypotheses and original sentences). For "1-to-1", NTS produces more simplifications, diverging more from the original sentences.

SENTENCE-LEVEL OPERATIONS. Interestingly, even though Moses and Nematus are trained on the same data, they differ substantially with respect to what they can learn. This is demonstrated by an automatic inspection we conducted on the simplifications produced by both systems trained over all types of sentence alignments, i.e. including sentence splits and joins.

	Moses		Nematus	
Operation	Count	%	Count	%
Identical	25,882	93.03	10,906	39.20
1-to-1	1,920	6.90	15,428	55.45
Split	14	0.05	354	1.27
Join	4	0.01	1,132	4.07

Table 4.2: Count and proportion of instances affected by each type of simplification transformation performed by Moses and Nematus.

Table 4.2 reports the count and proportion of instances in the test set representing types of sentence-level transformation between the original and simplified sentence. It can be noted that Moses is much more conservative than Nematus and simply tends to copy the original as the output ("Identical" cases). However, as the majority (57%) of aligned sentences in the professional Newsela simplifications are edited, we do not consider copying a valid "simplification" in most cases. Note also that Moses displays an excessively high BLEU score between the *original* and hypothesis sentences (98.77), while the similarity between the original and reference sentences is much lower (71.57).

Manually inspecting some of the simplifications made, we find that when it comes to sentence splits, both MT-based simplifiers seem to be able to perform this type of transformation in an accurate way. However, the proportion of such cases is very low (0.05% and 1.27% for Moses and Nematus, respectively) compared to the proportion in the gold data (13.5%) of the sentence pairs contain at least one split.

Moses only joins sentences in four cases, but these are all spurious instances where a period is incorrectly removed. Nematus is more successful at learning this type of operation. In most cases, it discards entire clauses that contain less relevant content. For example, it simplifies the sentence "*Lincoln often cried in public and recited sad poetry, according to Joshusa Wolf Shenk, who wrote a book called Lincoln's Melancholy*" to "*Lincoln often cried in public and recited sad poetry*". We also find a few examples where the content that is not discarded is rewritten to some extent, mostly for grammaticality. The Nematus simplification of "*Frank was what the instructors called a 'rock star'; he emerged as a leader who worked hard to keep the group together*" onto "*Frank was a leader who worked hard to keep the group together* " is a good example of that.

When it comes to 1-to-1 transformations, which can include a number of different operations (see §4.4), most transformations made by Nematus consist of segment deletions, some of which are paired with localized segment rewritings. As for Moses, most 1-to-1 outputs are

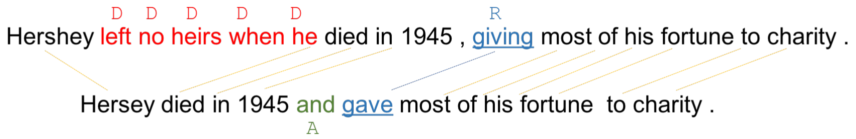

D D D D D
R
Hershey left no heirs when he died in 1945 , giving most of his fortune to charity .
Hersey died in 1945 and gave most of his fortune to charity .
A

Figure 4.1: Example of automatic labeling based on word alignments between an original (top) and a simplified (bottom) sentence in the Newsela corpus. Unaligned words on the original side receive label 'D' (DELETE), while words that are aligned to a different form receive 'R' (REPLACE). Aligned words without an explicit label receive a 'C' label (COPY). Sentences are from the Newsela Article Corpus.

identical to the original except for a few spurious typographic and punctuation changes. Because of that, Nematus simplifications are in average four tokens shorter than both complex originals and Moses simplifications.

A strong limitation of both models is their inability to address lexical complexity, performing very few lexical replacements. Most of the sentences that are lexically simplified have only one word replaced by another that does not preserve its original meaning. Take, for example, the word *clears* in the sentence *"It clears the way for troops on the ground with its huge bullets"*, which was replaced by *gathers* by Nematus, and the word *agribusiness*, which was replaced by *offering* by Moses in sentence *"Older brother Nate has taken college courses on livestock raising and agribusiness"*. Some of these issues become more evident in the human evaluation we performed comparing both end-to-end systems to our proposed approach (§4.5.2).

4.4 SIMPLIFICATION VIA SEQUENCE LABELING

Our approach to TS differs from translation-based models by explicitly predicting a set of operations to be applied at different positions in a complex sentence. Concretely, we tackle simplification as a sequence labeling problem, predicting operations at the token level and applying them downstream. As there are no high-quality and large-scale resources from which such operation sequences could be learned, we first generate training data as explained below.[14]

[14] For the experiments with the proposed TS approach, only 1-to-1 alignments are suitable. It is indeed not realistic to expect that complex operations that involve significant structural changes (e.g., splitting or joining sentences) could be modeled using sequence labeling approaches. For such complex operations, we believe explicitly representing the sentences' syntactic structures and learning abstract syntactic transformation rules (e.g. as in Woodsend and Lapata (2011) or Feblowitz and Kauchak (2013)) would be more advisable. However, we note that, as previously shown, translation-based end-to-end approaches also fail to learn such complex operations.

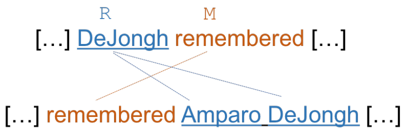


Figure 4.2: Example of automatic annotation for label MOVE ('M'). Sentences are from the Newsela Article Corpus.

4.4.1 *Generating Training Data*

Given 1-to-1 sentence pairs, our method for data generation identifies deletions, additions, substitutions, rewrites (replacing or adding non-content words), and reorderings performed between sentences pairs.

AUTOMATIC OPERATION ANNOTATION. The annotation process uses the following set of operation labels: DELETE (D), REPLACE (R), and MOVE (M) in the original (source) sentence; ADD (A) in the simplified sentence; and REWRITE (RW) in both.[15]

We first generate word alignments between the original and simplified sentences using the aligner by Sultan et al. (2014). Based on these alignments, we perform a word-level annotation for labels DELETE and REPLACE. Our heuristics are that if two words are aligned and are not an exact match, then the corresponding label is REPLACE. If a word in the original sentence is not aligned, it must be a DELETE, and if a word in the simplified sentence is not aligned, it is an ADD. In any other case, the word receives label C (COPY) or O (not part of a simplification operation) in the original or simplified sentence, respectively. For details, see Algorithm 1 in the supplementary material. Figure 4.1 presents an example for our automatic labeling approach. We consider REWRITE labels as special cases of REPLACE where the words involved are isolated (not in a group of same operation labels) and belong to a list of non-content words.

Finally, we label reorderings (MOVE) by determining if the relative index of a word (considering preceding or following deletions and additions) in the original sentence changes in the simplified one (Algorithm 2). See Figure 4.2 for an example. Words or phrases that are kept, replaced or rewritten, may be subject to reorderings, such that a token may have more than one label (e.g. REPLACE and MOVE). For that, we extend the set of operations by the compound operations REPLACE+MOVE (RM) and REWRITE+MOVE (RWM).

EVALUATION OF AUTOMATIC LABELS. To test our algorithms, we compare their output to manual annotations for 100 sentences from level pair 0-1 of the Newsela corpus. The manual annotations were

[15] Target-side annotations serve for analysis; they are ignored in our experiments as they are unavailable at test time.

Label	Prec.	Rec.	F_1	Support
A	0.66	0.92	0.77	261
D	0.76	0.90	0.82	371
M	0.17	0.92	0.28	24
R	0.70	0.39	0.50	71
RM	0.22	0.33	0.27	12
RW	0.24	0.07	0.11	57
RWM	0.00	0.00	0.00	6
C	0.99	0.94	0.96	1932
O	0.99	0.95	0.97	2112
avg / total	0.92	0.92	0.92	4846

Table 4.3: Per-label performance of automatic annotation of operations.

	Automatically Annotated								
	A	D	M	R	RM	RW	RWM	C	O
A	240	0	0	0	0	2	0	0	19
D	15	333	8	4	5	1	1	4	0
M	0	1	22	0	0	0	0	1	0
R	0	33	0	28	6	0	0	4	0
RM	0	8	0	0	4	0	0	0	0
RW	3	31	4	7	2	4	0	6	0
RWM	0	6	0	0	0	0	0	0	0
C	0	24	98	1	1	1	0	1807	0
O	105	0	0	0	0	9	0	0	1998

Table 4.4: Confusion matrix of true (rows) and automatically annotated (columns) operations on the manually annotated data.

performed by four proficient English speakers. For 30 of those sentences, we calculated the pairwise inter-annotator agreement between annotators, yielding an average kappa value of 0.57. We obtain an accuracy of 0.92 for all labels, and a micro-averaged F_1 score of 0.70 for all positive labels (i.e. excluding 'C' and 'O'). Table 4.3 presents details on the performance of our annotation algorithms over the identified operations. Of the positive labels, the algorithms annotate most accurately additions and deletions. According to the confusion matrix in Table 4.4, the relatively low ability of capturing replacements is due to labeling them as deletions. This is mainly caused by word miss-alignments and by parser errors that our heuristics cannot recover from. The same logic applies for labels replace+move and rewrite+move. We are also able to capture most movements (high recall), but our reordering heuristic still requires improvement.

We refer to these automatically generated labels as **silver labels**. As we describe in the next sections, the corpus annotated with these labels will be used to train our sequence labeling approach, eliminating the need for costly human-annotated data (i.e. gold labels). As a second way of evaluating the quality of our automatic labeling, we use these silver labels in a semi-oracle trial where we apply the actual simplification operations as given in the annotated corpus. In other words, we simply take the automatic labels as true and use the alignments between original and simplified words to apply the actual operations. This is what we refer to as **silver operations** in Table 4.1. Using the automatic labeling would lead to much more accurate and less conservative simplifications than all translation-based approaches: it achieves the highest SARI and BLEU scores, and the lowest rate of copied input sentences among all systems tested using the 1-to-1 alignments. Therefore, the challenges now are (i) to predict such labels (§4.5.1), and (ii) to devise high-performing TS modules to apply simplification operations for each type of label (§4.4.2).

4.4.2 *Application of Operations*

For our experiments (§4.5), we consider two of the operations that our algorithms can identify with high precision: DELETE and REPLACE.[16] Applying deletions is straightforward and amounts to simply omitting the respective token when generating the hypothesis sentence. For the REPLACE operation, we use the supervised Lexical Simplification approach of Paetzold and Specia (2017c). Their simplifier generates candidate substitutions for target words using parallel complex-to-simple corpora and retrofitted context-aware word embedding models, selects the ones that fit the context of the target word through the unsupervised boundary ranking approach, then ranks candidates using a supervised neural ranking model trained over manually annotated simplifications. It also performs a final confidence check step: the target is only replaced by the highest ranking candidate if the trigram probability of two words preceding the target is higher for the candidate.

4.5 EXPERIMENTS

Based on the automatic annotation procedure outlined above, we generate sequence annotations of 1-to-1 simplification operations in the

[16] We focus on this subset of operations since we currently lack good models to apply to the remaining operations. ADD, for example, would presume access to an external resource such as a knowledge base that would serve as a basis for inferring added content (which is oftentimes background information, for example an explanation that a certain person has a certain function). The results we obtain can thus be viewed as a lower bound on the simplification quality that can be expected from a model that integrates other operations.

Label	Prec.	Rec.	F_1	Support
D	.30	.49	.37	58,692
M	.21	.16	.18	29,719
R	.13	.34	.19	7,208
RM	.00	.00	.00	2,817
RW	.14	.07	.10	646
RWM	.00	.00	.00	141
C	.68	.51	.58	154,481
avg / total	.51	.45	.47	253,704

Table 4.5: Per-label performance of automatic operation prediction with the LSTM model.

Newsela corpus. On this data, we explore the questions (i) whether we can predict simplification operations to be performed on unseen data, and (ii) to what degree the prediction of these operations allows us to generate good simplifications.

4.5.1 *Prediction of Simplification Operations*

To predict simplification operations for each input word, we train a bidirectional recurrent neural network, with an initial embedding layer of size 300 and two hidden LSTM (Long-Short Term Memory) layers of size 100. The training is done using Keras (Chollet, 2015), with a batch size of 64, categorical cross-entropy loss and a dropout rate of 0.2 after the hidden layers. We optimize the model with Adagrad (Duchi et al., 2011). We monitor the tagging accuracy on held-out development data and employ early stopping when the development loss increases. We repeat this process ten times with random initializations and select the best model based on development set accuracy.

Table 4.5 shows that the LSTM model does not predict the silver labels very well. In particular, the model is relatively conservative with respect to the prediction of simplification operations, and tends to overpredict the majority class (i.e., to copy a token).[17] DELETE is the operation that our model predicts best. Table 4.6 shows the relative confusion of predicted operations versus the silver labels, and confirms that the main error type of our system is to keep a token rather than performing some simplification operation on it. We also see a tendency for other operations to be predicted as deletions.

The results in the lower part of Table 4.1 (“Predicted operations (1-to-1)"), however, show that even though the operation predictions are

[17] By weighting the loss function by the ground truth class support at each timestep, we were able to alleviate the effect of a predominant majority class to some degree.

	Predicted						
	D	M	R	RM	RW	RWM	C
D	.49	.06	.07	.00	.00	.00	.38
M	.41	.16	.05	.00	.00	.00	.38
R	.23	.05	.34	.00	.00	.00	.38
RM	.32	.09	.21	.00	.00	.00	.38
RW	.38	.00	.00	.00	.07	.00	.54
RWM	.62	.03	.00	.00	.04	.00	.32
C	.33	.09	.06	.00	.00	.00	.51

Table 4.6: Confusion matrix of true (rows) and predicted (columns) operations on the test data.

far from the silver labels, our system is able to generate simple output by only applying the DELETE and REPLACE operations. In particular, our method achieves a better SARI score than all the baseline systems on the 1-to-1 alignments. As we consider the extrinsic evaluation of the final TS results to be more indicative of the quality of our model than its intrinsic evaluation in the sequence labeling task, we view this as a positive result.

4.5.2 *Human Evaluation*

We finally conduct a human evaluation of 100 simplifications produced by five simplifiers:

- The experts' **Reference** simplification.
- The **Moses** simplifier (1-to-1).
- The **Nematus** simplifier (1-to-1).
- The **NTS** simplifier (1-to-1).
- Our **Sequence Labeling (SL)** simplifier.

Human evaluators (four NLP experts) are given the original sentence and the simplification in each of the above versions, and are asked to judge each of them with respect to their grammaticality (G), meaning preservation (M) and simplicity (S), using a Likert scale between 1 (worst) and 5 (best) for each aspect. We define "simplicity" as the extent to which the sentence was simpler than the original and thus easier to understand. A control set of 20 sentences is evaluated by all annotators in order to compute inter-annotator agreement.

	G	M	S
Reference	5.00±0.0	4.45±0.9	2.70±1.3
SL	4.16±1.0	3.91±1.1	1.66±0.9
Nematus	4.49±0.9	3.99±1.2	1.46±0.9
Moses	4.98±0.2	4.99±0.1	1.14±0.4
NTS	4.75±0.6	4.08±1.3	1.53±1.0
Fleiss' Kappa	0.372	0.457	0.342

Table 4.7: Average scores and standard deviation for grammaticality (G), meaning preservation (M) and simplicity (S) for the systems evaluated. The last row shows the inter-annotator agreement scores in terms of Fleiss' Kappa.

Table 4.7 illustrates the average scores and standard deviations obtained by each system according to each criterion. As expected, the Moses simplifier obtains the highest grammaticality and meaning preservation scores, but the lowest simplicity scores, given that it tends to merely reproduce the input. Although Nematus and NTS manage to obtain slightly higher simplification scores, they still average very close to the lower end of the simplicity scale. Our SL approach, in turn, shows significantly higher simplicity scores than the other systems (according to a t-test with $p < 0.05$). Its less conservative edits, however, may in some cases come at the cost of lower scores for grammaticality and meaning preservation. The last row in Table 4.7 shows the values of inter-annotator agreement in terms of Fleiss' Kappa for each evaluation aspect. Table 4.8 exemplifies some of the sentences for which our system was rated better and worse than the baselines. It is important to mention that, although the first two reference simplifications in Table 4.8 feature only minor punctuation changes, only 2,538 references (0.8%) in the dataset are of this type.

4.6 CONCLUSIONS AND FURTHER WORK

We presented a novel approach to sentence simplification that uses automatically labeled training data from a large simplification corpus. Based on this annotated corpus, we devise a sequence labeling approach to text simplification that predicts simplification operations for individual words in the original sentence. Specific modules are then triggered to deal with each predicted operation. The experiments reported here cover only deletions and lexical substitutions as operations.

Our approach has several theoretical advantages over end-to-end translation models, including easier interpretability of the types of simplification learned, as well as the possibility for late decoding for

	SL better than other Moses, Nematus and NTS
O	Kyarra Garrett has learned how to take blood pressure and perform CPR – and she is not even out of high school yet.
R	Kyarra Garrett has learned how to take blood pressure and perform CPR, and she is not even out of high school yet.
M	*Kyarra Garrett has learned how to take blood pressure and perform CPR – and she is not even out of high school yet.*
N	*UNK Garrett loves out to take blood pressure and perform, and she is not even out of high school yet.*
T	*Chance Garrett has learned how to take blood pressure.*
L	*Kyarra Garrett has learned how to take blood pressure and perform CPR.*
O	in her mind she stops at particular locations to pick up the correct cookie crumbs.
R	in her mind, she stops at particular locations to pick up the correct cookie crumbs.
M	*in her mind she stops at particular locations to pick up the correct cookie crumbs.*
N	*she stops at particular locations to pick up the correct cookie UNK.*
T	*in her mind she stops at particular locations to pick up the correct cookie momentum.*
L	*in her mind she stops at particular areas to pick up cookie crumbs.*
	SL worse than Moses, Nematus or NTS
O	despite the limitations, Palestinian cooking is not without its fans.
R	despite the limitations, Palestinian cooking has its fans.
M	*despite the limitations, Palestinian cooking is not without its fans.*
N	*Palestinian cooking is not without its fans.*
T	*even Palestinian cooking is not without its fans.*
L	*despite the limitations, Palestinian cooking is not without its fans.*
O	"we always thought there has to be a more efficient way of doing this," Zach Fiene said.
R	he said he always thought there had to be a better way of doing it.
M	*"we always thought there has to be a more efficient way of doing this," Zach Fiene said.*
N	*"we always thought there has to be a more efficient way of doing this," said Zach Ghani, who is the 18-year-old said.*
T	*Zach Fiene said there has to be a more efficient way of doing this.*
L	*"we always thought there has to be more efficient way doing this said.*

Table 4.8: Example including original (O) and reference (R) sentences from the Newsela Article Corpus, and outputs generated by Moses (M), Nematus (N), NTS (T) and our sequence labeling approach (**L**).

adaptive simplification. In practical terms, we showed that our system outperforms translation-based approaches on a number of metrics and overcomes the problems of excessive repetition of the original content.

According to human evaluation, our system achieves higher simplicity scores than the baseline systems, although this comes at the cost of slightly lower meaning preservation and grammaticality. We hypothesize that some of the problematic cases stem from not realizing the addition operation. In general, our approach will likely profit from good models for the remaining operations, especially those that can also operate on spans of several tokens, making research on such models a natural direction for further work.

ACKNOWLEDGEMENTS

This work was partly supported by the EC project SIMPATICO (H2020-EURO-6-2015, grant number 692819).

Part III

PERSONALIZED LEXICAL SIMPLIFICATION

5

PREDICTING MISREADINGS FROM GAZE IN CHILDREN WITH READING DIFFICULTIES

ABSTRACT

We present the first work on predicting reading mistakes in children with reading difficulties based on eye-tracking data from real-world reading teaching. Our approach employs several linguistic and gaze-based features to inform an ensemble of different classifiers, including multi-task learning models that let us transfer knowledge about individual readers to attain better predictions. Notably, the data we use in this work stems from noisy readings *in the wild*, outside of controlled lab conditions. Our experiments show that despite the noise and despite the small fraction of misreadings, gaze data improves the performance more than any other feature group and our models achieve good performance. We further show that gaze patterns for misread words do not fully generalize across readers, but that we can transfer some knowledge between readers using multitask learning at least in some cases. Applications of our models include partial automation of reading assessment as well as personalized text simplification.

5.1 INTRODUCTION

Reading disabilities are impairments affecting individuals' access to written sources, with downstream effects such as low self-confidence in the classroom and limited access to higher education. Dyslexia, for instance, while being highly prevalent with estimates reaching up to 17.5% of the entire population of the U.S. (Interagency Committee on Learning Disabilities, 1987), often goes undiagnosed, such that unattributed weaknesses in reading comprehension further intimidate affected persons. Due to these severe and broad-ranging impacts of reading difficulties, many governments have implemented early screening tests for dyslexia and other reading difficulties and provide special training and assistance for struggling readers throughout the educational system and into adulthood.

In Denmark, for example, such programs provide children with specialist training through focused multi-week reading courses in one-on-one or small group settings. Still, the specialized teachers can only attend to one student at a time when closely monitoring their reading, and the quality of any analysis is strictly limited by the

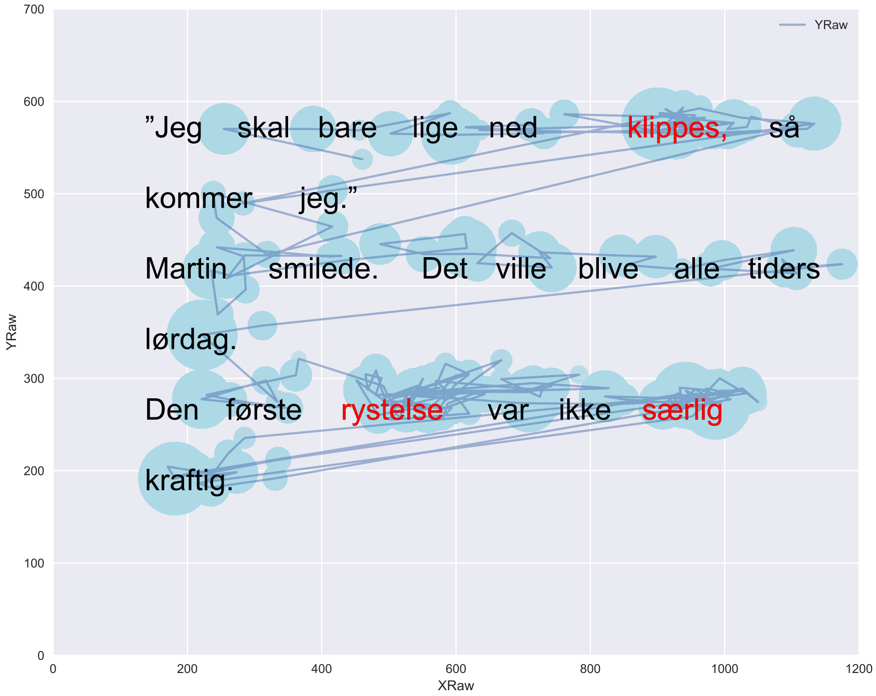


Figure 5.1: Scanpath and fixations (blue circles) when reading a sentence. This particularly clear example from our dataset shows extended processing time for misread words (marked in red).

human observer's processing “bandwidth" while attending the live reading.

As a possible mitigation, advances in eye-tracking technology–in particular the increased availability of eye trackers–have made it possible to reliably record children's gaze during reading, both allowing teachers to attend to their students' reading post-hoc as well as providing additional insight into reading strategies based on gaze, including the development of these strategies over time. For the teacher to track and keep records of reading mistakes (henceforth referred to as *misreadings*), however, the students are still required to read out loud, and the teacher has to review the entire reading and annotate for misreadings.

In this work, we investigate to what extent we can predict misreadings from gaze patterns for individual words. While the aim is not to fully automate reading reviews, being able to successfully predict misreadings from gaze data can be part of a semi-automatic system for reading quality assessment and increase teacher efficiency by pointing out potential misreadings for closer review.

Another motivation for this work comes from text simplification, in particular from the observation that individuals' highly specific reading strengths and weaknesses require text simplification models to be customized to specific users in order to unfold their full potential and truly be helpful. Predicting misreadings in concrete reading scenarios and based on individual gaze patterns can be used as a first step in the typical lexical simplification pipeline (Shardlow, 2014b).[18] This task, known as complex word identification, has received a considerable amount of attention in the literature, but has exclusively been approached in a user-agnostic fashion.

The data used in this study are gaze recordings of children with reading difficulties, reading Danish texts assigned by their reading teacher as part of their reading intervention. The recordings stem from EyeJustRead, an eye-tracking based software used in special reading intervention in Danish schools.[19] In Section 5.3, we discuss further aspects of the treatment of gaze data in general and the collection of the data used in this study in particular.

While the difficulty of processing a word is undoubtedly reflected in the fixation time on that word (Rayner et al., 1989), many other factors affect fixation durations, the most prominent being word length and word frequency, but also predictability and relative position in sentence have strong effects–see Figure 5.1 for a particularly clear example from our dataset. Notably, almost all analyses of eye-tracking reading data use data collected in research laboratories, where these–otherwise confounding–factors can be controlled for. We show that we can perform reasonable misreading detection on real-world eye tracking data, including a limited number of textual features to control for these factors.

CONTRIBUTIONS a) We present the first work on the automatic detection of misreadings based on gaze patterns of children with reading difficulties. b) This is, to the best of our knowledge, the first attempt at modeling noisy, real-world eye-tracking data from readers. c) We also present, to the best of our knowledge, the first published results using a multi-task learning setup to transfer knowledge between individual readers for personalized, complex word identification.

5.2 RELATED WORK

Our work is a special case of complex word identification, a task that has recently received a significant amount of interest, including two shared tasks (Paetzold and Specia, 2016b; Yimam et al., 2018). The most successful approaches to these tasks had in common that they employed ensembles of classifiers that learned from a number of semantic and psycholinguistic features. Note however, that these previ-

[18] While today it may hardly sound plausible to equip each laptop with an eye-tracker in order to track people's reading, further technological advances may well make this possible in the future. Recent development in eye-tracking technology has taken it from expensive research equipment to a gaming interface with a price point as low as $100.

[19] http://www.eyejustread.com

ous approaches to complex word identification aimed at developing generic models that took no account of any specifics of a certain user.

Children's eye movements during reading are not as well-studied as adults', and previous studies typically analyze data collected in experiments designed for research. The overall established observations with regards to reading development are: older children have shorter fixation durations, fewer fixations and fewer regressions. They have a higher skipping probability and also higher saccade amplitude. See Blythe and Joseph (2011) for a review. It is not conclusive whether these variations follow chronological age or their increased reading proficiency. Regardless of the underlying cause, due to the observed systematic differences, the standard procedure is to control as closely as possible for age and reading proficiency level when designing reading experiments.

There are several psycholinguistic studies that show that also in children, the typicality and plausibility of sentences (Joseph et al., 2008) as well as temporary sentence ambiguity (Traxler, 2002) can be traced in eye movements, suggesting that also other types of comprehension difficulties are reflected in the reading patterns.

Using gaze data to augment models is a recent addition to NLP. Previous approaches that have used gaze data in the context of natural language processing include the work of Barrett et al. (2016), who aim to improve part-of-speech induction with gaze features, Klerke et al. (2016), where gaze data is used as an auxiliary task in sentence compression, and Klerke et al. (2015b), where gaze data is used to evaluate the output of machine translation. The most related work is Klerke et al. (2015a) and Gonzalez-Garduño and Søgaard (2017). Klerke et al. (2015a) compared gaze from reading original, manually compressed, and automatically compressed sentences. They found that the proportion of regressions to previously read text is sensitive to the differences in human- and computer-induced complexity. Gonzalez-Garduño and Søgaard (2017) show that text readability prediction improves significantly from hard parameter sharing when models try to predict word-based gaze features in a multi-task-learning setup. All of these works, however, use gaze data that was collected under laboratory conditions from skilled, adult readers.

5.3 GAZE DATA

In eye-tracking studies, gaze data is normally sampled under experimental circumstances, where e.g. instructions, location, environment, lighting, participant sampling, textual features, order, duration etc. are controlled for. Our real-world data, on the contrary, lacks all of these controls. While in controlled, cognitive psychology experiments, fixation durations have proven to systematically correlate with cognitive load (see Rayner (1998) for a review), eye movements from-real world applications have been largely understudied, and specific findings from the literature on controlled data may not apply here or may be swamped by extraneous factors. Further, the often-used statistical tests of significant differences between gaze patterns lose some of their legitimacy when data is retrieved under noisy conditions.

5.3.1 *Data collection and preprocessing*

The data we use in this work is collected in Danish schools using commercial software specifically developed to record and track children's reading development. The system records the eye movements and voice while the children are reading aloud. The teacher can afterwards replay the reading along with the recorded eye movements. The software performs some low-level eye-movement analyses to help the teacher understand how the child processes the text. The teacher can mark which words are erroneously read by the child and later access this and other basic statistics about the reading – see Klerke et al. (2018) for a workflow description. The genre is children's fiction books and the children read contextualized, running text.

As the data is fairly noisy compared to data from laboratory-based eye tracking experiments, we perform thorough cleaning before running any experiments. This cleaning procedure is described below. Table 5.1 contains a summary of the dataset sizes after each cleaning step. Before any cleaning is performed, the dataset contains 369 reading sessions from 95 unique readers. In total it has 3,161 read pages.

HELP WORD ACTIVATED ON PAGE We start by removing all pages where the reader activated the help word function, which dynamically isolates and enlarges a single word on the screen. This dynamic display generates a series of eye movements that do not resemble typical reading activity. This step removes 94 pages.

FIXATION DETECTION We pre-process the raw gaze data by first detecting fixations using a custom implementation of the algorithm of Nyström and Holmqvist (2010). We remove fixations shorter than 40ms and longer than 1.5s.[20] For the calculation of gaze features (see below), we further discard all data points that are not detected as a fixation on text (but instead on images or blank parts of the page). We remove 19 pages where we do not have any fixations on text (e.g. due to the reader just browsing through a book or because of technical issues).

[20] Removing short fixations also removes the majority of blinks which presents as a sudden downward-upward pattern of saccades separated by a pause in the signal or a short, falsely detected fixation.

Cleaning step	Reading sessions	Unique readers	Read pages	Read words	Misreadings
No cleaning	369	95	3161	73,965	644
Help word activated	366	95	3067	71,911	619
Fixation detection	366	95	3048	64,191	613
Bad calibration	335	87	2865	56,166	565
Marked by teacher	83	44	405	8,681	565

Table 5.1: Dataset size after each cleaning step

BAD CALIBRATION Prior to reading, the student is prompted to calibrate the eye tracker. In the data used in this study, most reading sessions (91%) attain the best calibration score on a five-point scale, while 6% miss a calibration score. The remaining 3% do not have the best calibration score. We remove everything but the 91% with the best calibration score.

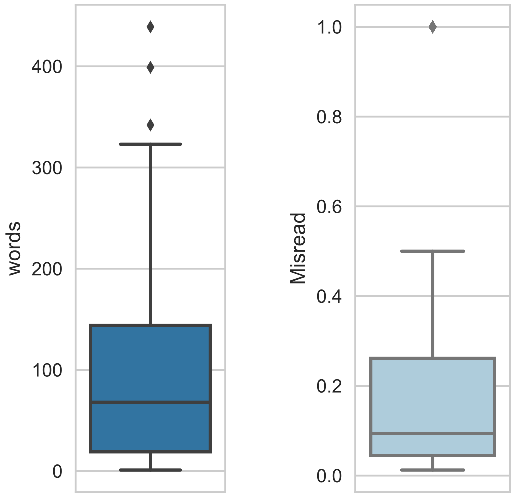
Only parts of the readings have been reviewed and marked for misreadings by a teacher. However, whether a teacher reviewed a reading or not is not explicitly encoded in the data. Thus, if there are no marked misreadings in some session, we do not know whether this is because this reading was not reviewed or because there actually were no errors. We therefore remove all readings without any marked misreadings, as well as any data before the first marked misreading and after the last marked misreading within marked sessions, assuming that everything between these two points has been marked. Twelve cleaned reading sessions only consist of one misread word – everything before and after was removed. See Figure 5.2(a) for an overview of the distribution of number of words per reading after this cleaning step. This leaves us with the subset of the readings that posed most problems for the subjects. Figure 5.2(b) shows the distribution of misread words in the cleaned dataset. It is worth noting that since this is not controlled, experimental data, “misread" is not necessarily interpreted equally by all teachers, or even consistently across markings from the same teacher, due to the lack of an annotation protocol. We assume that “misread" means that the pronounced word deviates substantially from the written word. Ultimately, we retain 83 reading sessions from 44 readers with at least one misread word.

5.3.1.1 *Apparatus*

The eye tracker used is a Tobii Eye Tracker 4C with a sample rate of 90 Hz. It is an affordable, consumer eye tracker targeted at gaming. The laptop computers to which the trackers are attached, and which run the software, are provided by the different institutions and vary. Screen resolution is locked by the eye tracker software to 1366 x 768, and most systems reportedly run on a 14"–15.6" monitor. The font size is 50pt, which is equivalent to approximately 6mm x-height. Distance between baselines was approximately 18mm with the most commonly used font–otherwise 24mm.

5.3.1.2 *Subjects*

The cleaned dataset contains 44 unique readers with different reading durations. Readers are probably between 5 and 15 years old, which is the official age of students in the Danish schools, but we do not know their exact ages. To control for reading proficiency, we include the texts' readability scores as a feature in all experiments. All stu-



(a) Words/session (b) Misreading/session

Figure 5.2: Distributions of total number of words and misreading ratios per session after cleaning.

dents receive extra reading classes, because they struggle with reading. Many of them are probably dyslexic, but we do not have access to this information. Because this is not experimental data, the students will have received different instructions from the teachers. We do not know if they picked the text themselves or for how long they read prior to each recording. They are not necessarily alone in the room, but it is a fair assumption that they all make an effort to read correctly because they are recorded. The data comes from a number of different systems that we were informed is in the range between 10 and 20, but the actual number of schools and teachers is unknown to us. All children and their parents gave consent that the anonymized eye-tracking data may be used for this research.

5.3.2 *Features*

Reading patterns have been shown to be influenced by a number of factors, including textual features and the instructions given to a reader, such as encouraging a specific reading strategy. Readers, or different groups of readers, furthermore display individual reading styles which affect the eye movements (Benfatto et al., 2016). Other factors include the reader's individual skill level, cognitive abilities and mood, among others.

We extract a number of gaze features that have been associated with processing load. Some of our gaze features directly reflect the processing load associated with a word, especially the two correlated measures *total fixation duration* and *number of re-fixations*, but also the *mean fixation duration*. Some gaze features are included to account for preview effects (whether the next or previous word was fixated) as well as the scan path immediately surrounding the word. We split the gaze features into two groups: GAZE (W) for features directly associated with word-level processing and GAZE (C) for features associated with the eye movements on the immediate context of the word. All features are scaled to the $[-1, 1]$ interval.

We further extract a number of basic features that are known to affect gaze features and thus need to be controlled for. These include word length and word frequency (Hyönä and Olson, 1995), but also position in sentence (Rayner et al., 2000) and position on the page have shown to affect reading for adults. We also include a range of linguistic features that we expect to describe word difficulty. All features and feature groups are listed in Table 5.2 and described below.

GAZE FEATURES During reading, the reader performs a series of stable fixations of a couple of hundred milliseconds duration on average. Between fixations, the eyes perform rapid, targeted movements, called *saccades*. All gaze features are computed on the word level and use the application's definition of the area of interest surrounding each word.

For gaze duration, we extract both late and early processing measures. Late measure such as total *fixation duration* and *number of refixations* reflect late syntactic and semantic processing in skilled adult reading (Rayner et al., 1989). For children with reading difficulties, we assume these measures to likely reflect processing difficulty.

For the first three passes over a word, we also extract the direction and the word distance of both the ingoing and outgoing saccade.[21] These six features are expected to map the activity around the word and, for example, show whether some word was part of sequential, forward reading or occurred in a series of erratic saccades.

Four features indicate the *landing positions* of fixations in four equally-sized parts of the display width of a word. This captures whether a word, for instance, has three fixations on the last quarter of its display width, which would be atypical and suggest that the reader is struggling with the ending of this word. We further explicitly encode the landing position of the first and last fixation. Note that because of the anatomy of the eye, eye tracking can never be pixel-accurate, but has at least 2° inaccuracy. For short words (or words printed very small, which does not apply for this study) these features may be misleading.

[21] As we removed everything that was not a fixation on text before calculating the gaze features, intermediary non-text fixations may have occurred between text fixations, such as image fixations. We count the last/next fixated *word*. For example, if a word has index 5, and the first pass incoming saccade is from word index 4, we get a feature value of -1 for first pass ingoing.

Basic	Gaze on Word (W)
Is bold	Number of fixations on word
Is italic	First fixation duration
Is lowercase	Mean fixation duration
Is uppercase	Total fixation duration
Has punctuation	Count of passes over the word
Line index on page	Left pupil size
Word index on line	Right pupil size
Page number	Refixation counts
Position in sentence (relative)	Fixations in first quarter count
Position in sentence (absolute)	Fixations in second quarter count
Sentence length (characters)	Fixations in third quarter count
Sentence length (words)	Fixations in fourth quarter count
Word index	Relative landing position of first fixation
Sentence index	Relative landing position of last fixation
Word length (characters)	Average character index of fixations
Gaze in Context (C)	Linguistic
1st pass ingoing saccade dist. and dir.	LIX score for entire text
1st pass outgoing saccade dist. and dir.	Previous occurrences of word stem in text
2nd pass ingoing saccade dist. and dir.	Previous occurrences of word type in text
2nd pass outgoing saccade dist. and dir.	Vowel count
3rd pass ingoing saccade dist. and dir.	Character perplexity
3rd pass outgoing saccade dist. and dir.	Word frequency
Next word fixated	Universal POS tag
Previous word fixated	

Table 5.2: Overview of the feature groups used in the experiments.

The data also provides pupil sizes for both eyes. It is well known that the pupil dilates as response to external lighting factors, but there is also evidence that the pupil systematically–but on a much smaller scale–dilates as a response to mental state, emotions or concentration (Beatty, Lucero-Wagoner, et al., 2000). In an experiment collecting pupil size, one would control lighting, which was not possible in the present scenario. For all pupil measures, we subtracted the same side mean of the reading session. We confirmed that all changes larger than 0.6 times the mean were captured when removing short fixations, as they may be caused by the tracker mistaking eyelashes for pupils during blinks.

BASIC FEATURES The basic features span 16 textual and presentational features that are either directly accessible via the system or easily obtainable. They are included in all our experiments and serve as control features for the gaze features because we expect them to explain some of the variance in the gaze features, e.g. reading changes over the course of a line and the course of a sentence (Just and Carpenter, 1980). We further encode the line number a word is located in on a page, as well as its position in that line.

LINGUISTIC FEATURES The linguistic features include the absolute vowel count, which in Danish is highly correlated with the number of syllables. Universal POS tags are obtained from the Danish Polyglot tagger.[22] We also include the provided *Läsbarhetsindex* (LIX) (Björnsson, 1968), a Swedish readability metric (commonly also applied to Danish) that considers the mean sentence length and the ratio of long words (more than 6 characters). The log word probability is estimated from a language model we train on the entire Danish Wikipedia (downloaded in November 2017) using KenLM (Heafield, 2011). Frequency affects processing load and thus fixation duration for adults as well as dyslexic and neurotypical Finnish children (Hyönä and Olson, 1995), but there is conflicting evidence as to whether text frequencies from adult text explain variance in children's eye movements (Blythe and Joseph, 2011). Character perplexity is estimated using a 5-gram character language model, also using KenLM on the Danish Wikipedia. The previous occurrence of stems and word types is included as reading time for low-frequency words has shown to decrease on later repeats in a text (Rayner et al., 1995). We use NLTK's snowball stemmer for Danish.

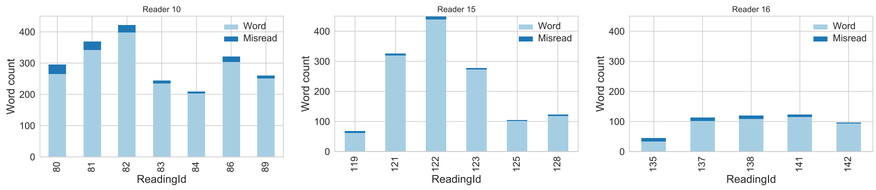
5.4 MODEL

In preliminary experiments, we observed that the relatively small overall amount of data, as well as the low fraction of positive in-

[22] http://polyglot.readthedocs.io

Feature group	F_1	
BASIC	18.78	†
+ GAZE (W)	40.50	*
+ GAZE (C)	18.49	†
+ LINGUISTIC	19.24	†
+ GAZE (W) + GAZE (C)	**41.19**	*
+ GAZE (W) + LINGUISTIC	41.08	*
+ GAZE (W) + LINGUISTIC	18.65	†
All features	40.42	*

Table 5.3: Performance across feature groups for Experiment 1. Scores are averaged F_1 over ten cross-validation folds. Using an independent t-test, * and † indicate results from ten cross validation rounds significantly different from BASIC and the best feature combination BASIC + GAZE(W) + GAZE(C), respectively.



(a) Reader 10

(b) Reader 15

(c) Reader 16

Figure 5.3: Words and misreading counts for readings of three readers in cross-user experiment

stances, caused significant variation between repeated random restarts of various classification algorithms. We thus approach the task of predicting misreadings from gaze with ensemble methods, training N classifiers independently on the same data and letting them vote on the instances in a held-out development set. Using this development set, we then optimize a threshold t, which is the fraction of the number of classifiers that need to cast a positive vote on an item before we accept it as such.

All of our ensembles consist of 10 random forest classifiers and 10 feed-forward neural networks. The random forests, in turn, consist of 100 trees that create splits based on Gini impurity (Breiman, 2001). The neural network models are implemented in Pytorch and trained with the Adam algorithm (Kingma and Ba, 2014), with an initial learning rate of $3 \cdot 10^{-4}$ and a dropout rate of 0.2 on the hidden layers, whose number and sizes we vary in our experiments. We fur-

UserId	Number of	Words per reading		Thereof misread	
	reading sessions	Mean	std.dev.	Mean	std.dev.
10	7	285.9	67.5	16.6	9.9
15	6	219.2	148.1	5.0	2.3
16	5	91.6	32.7	8.0	3.1

Table 5.4: Statistics of (misread) words in sessions for the three readers with most readings.

ther employ early stopping, monitoring the loss on the development set with a patience of 30 steps.

5.4.1 *Multi-task learning for cross-user knowledge transfer*

One of the central questions we investigate in this paper is to what degree gaze patterns for misread words vary between readers, and whether we can learn to transfer knowledge about predictors of misreadings between readers. We address these questions in the experiments reported in Section 5.5.2, for which we use a multi-task learning (MTL) model that employs hard parameter sharing. MTL has received significant attention in the natural language processing community over the past years (see Bjerva (2017a) for a review). One of the most intriguing properties of MTL is that it allows for the transfer of knowledge between different tasks and datasets, which has been investigated and exploited in a growing number of works (Klerke et al., 2016; Martínez Alonso and Plank, 2017; Bingel and Søgaard, 2017), including work on the identification of complex words (Bingel and Bjerva, 2018).

In this work, we view the different readers as different *tasks*, motivated by Bingel and Bjerva (2018), who interpret different languages as different tasks for cross-lingual complex word identification. We define a feed-forward neural network model with one output layer per reader, all of which are dense projections from a shared hidden layer. In this framework, each training step consists of flipping a coin to sample any of the tasks and retrieving a batch of training data for this task. This batch is then used to optimize both the shared and the respective task-specific parameters. For a detailed definition of the model, see Bingel and Bjerva (2018).

5.5 EXPERIMENTS

5.5.1 *Experiment 1: Across entire dataset*

As a first experiment, we investigate the performance of our models and the predictiveness of the individual feature groups through 10-fold cross validation across the entire dataset. At each fold, we reserve one tenth of the data for testing and another tenth to monitor validation loss of the network as the early stopping criterion.

Note that we split the data randomly and do not stratify the cross-validation splits in any way. In conjunction with the strong class imbalance, this means that we are likely to encounter very different class distributions across splits. This setup may generally lead to lower performance scores, likely with greater variance. However, this was a deliberate choice as we cannot assume a consistent class distribution across train and test set in the real world, or in fact hardly any prior knowledge with regards to class distribution in the test set. Random splitting also means that data from the same *reading* will likely be distributed across train and test partitions for a certain cross-validation iteration.

We perform a first baseline experiment with only the basic features that we list in Section 5.2. On top of this baseline feature set, we perform further experiments, incorporating all combinations over the other feature groups. The results we present in Table 5.3 are based on the best respective model architecture for each feature combination, evaluated via the average over validation splits.[23]

5.5.2 *Experiment 2: Cross-reader prediction*

WITHOUT READER'S OWN DATA In a second experiment, we are interested in how well our model can predict misreadings for specific readers. For this, we identify the three readers with most reading sessions and perform a range of experiments, testing our models on the readings of each of these readers after training them on all other data. We denote the three most active readers by their unique, anonymized IDs as they appear in the dataset: 10, 15 and 16. These readers have 7, 6 and 5 recorded and marked readings, respectively, and we present statistics on these readings in Table 5.4 and Figure 5.3. As in the previous experiment, we optimize our model through cross validation to tune hyperparameters and perform early stopping. We report test data results for the model with optimal validation performance in Figure 5.4, broken down into each reader's different sessions.

[23] To address the variation in input dimensionality as we consider different feature group combinations, we train models with different architectures: (i) a single hidden layer with 20 units, (ii) two hidden layers with 20 units each, and (iii) a single hidden layer with 40 units.

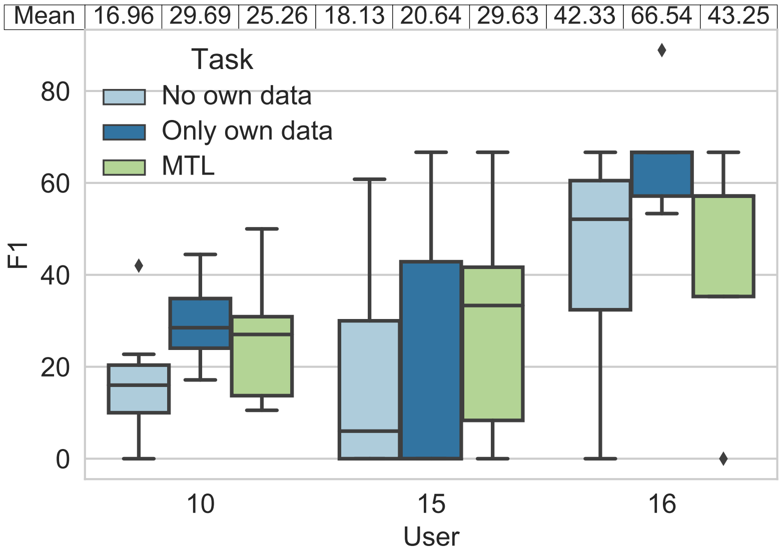


Figure 5.4: F_1 score distributions across test readings for each of the three readers with most sessions for three tasks.

learning from reader's own data Complementing the setup above, we now investigate how data from the same reader, but from different reading sessions, can inform our models. Therefore, we further perform cross-validation experiments across each reader's sessions. More concretely, for a reader with n marked readings, we perform n-fold cross validation, holding out one reading a time as a test set and another to monitor validation loss for early stopping of the neural model, while training on the remaining $n-2$ readings.

mtl As outlined in Section 5.4.1, we now view readers as tasks in an MTL model. For each of the three readers identified above and for each test reading, we train an ensemble whose neural MTL models define two outputs: one for the reader in question and one combined output for all other readers in the entire dataset. The random forest classifiers are trained on all remaining data except the held-out validation and test readings.

5.6 results and discussion

From Experiment 1, we observe that gaze features of the target word itself contribute strongly to model improvements over the baseline of textual features (see Table 5.3). Contextual gaze features and linguistic features do so to a lesser degree. The best feature group combination consists of the basic features and both gaze feature groups. Adding the linguistic features to this seems to slightly dilute the model.

The results from Experiment 2 in Figure 5.4 show that, at least for these three readers, there is a considerable degree of specificity attested in the reading patterns of misread words: in the scenario where we learn only from other users' gaze patterns (shown in light blue), performance is generally worse than for the other approaches. The high degree of reader specificity is also reflected in the comparison between learning just across a single user's readings and a multi-task setup that also considers other readers. Here, we observe that the former attains higher mean F_1 scores across readings for readers 10 and 16, although MTL is superior to the single-task setup for reader 15. Another observation is that misreadings can generally be predicted much better for reader 16 than for the other readers, which may in part be due to the higher ratio of misread words in these readings.

As especially our cross-reader experiments show, there is reason to believe that the manifestations of misreadings in gaze differ strongly between these readers. However, since we do not have information on the individual readers' age or general reading proficiency, we cannot confidently conclude whether the better stability of within-user experiments attested in Figure 5.4 is due to reader-specific idiosyncrasies or group-internal patterns (which would be supported by evidence that readers 10 and 16 were more atypical readers than others in the present dataset). We find some support for the latter hypothesis in literature describing children's reading development, which identifies a range of patterns common to young and low-proficiency readers. These patterns include longer and more frequent fixations, shorter saccadic amplitude and more regressions – all of which are also associated with comprehension difficulties, see Blythe and Joseph (2011) for a review. The presence of group-internal patterns is further supported by the observation that we are still able to successfully transfer knowledge about readings patterns between users in some cases, increasing performance for the readings of user 15.

One disadvantage of noisy, real-world data is that we do not know to what degree similarities and differences in the data, as well as our results, are influenced by chance, or whether they will generalize to other gaze data. The fact that many parameters are outside of our control and also outside of our knowledge means that we cannot describe certain biases in the data (such as age or reading skill) and consider them as causes for statistical variations in model performance.

5.7 CONCLUSION

This paper presented first work in the automatic prediction of reading errors in children with dyslexia and other reading difficulties using real-world gaze data. We showed that despite the noisy conditions under which this data was obtained, features we extract from the gaze patterns are predictive of reading mistakes among children. Besides the immediate application in automating some parts of reading teaching, this could be exploited in personalized text simplification, where gaze could be used as feedback to the system.

Our experiments further show that while gaze patterns for misreadings seem to be largely specific to individual readers or groups of readers, we can successfully use MTL to transfer knowledge between readers at least in some cases. Note also that we have very little knowledge of the age and general proficiency of specific readers, including those investigated in our MTL experiments, and we expect that our MTL approach can be much more successful between more similar readers.

ACKNOWLEDGEMENTS

The authors wish to thank the children using EyeJustRead, as well as their parents, for giving us permission to use their data. We are also grateful to Emil Juul Jacobsen and Janus Askø Madsen for supplying us with the data. We further acknowledge valuable comments by Anders Søgaard as well as by the anonymous reviewers. We acknowledge the support by Trygfonden.

6 LEXI: A TOOL FOR ADAPTIVE, PERSONALIZED TEXT SIMPLIFICATION

ABSTRACT

Most previous research in text simplification has aimed to develop generic solutions, assuming very homogeneous target audiences with consistent intra-group simplification needs. We argue that this assumption does not hold, and that instead we need to develop simplification systems that adapt to the individual needs of specific users. As a first step towards personalized simplification, we propose a framework for adaptive lexical simplification and introduce Lexi, a free open-source and easily extensible tool for adaptive, personalized text simplification. Lexi is easily installed as a browser extension, enabling easy access to the service for its users.

6.1 INTRODUCTION

Many a research paper on text simplification starts out by sketching the problem of text simplification as rewriting a text such that it becomes easier to read, changing or removing as little of its informational content as possible (Zhu et al., 2010; Coster and Kauchak, 2011c; De Belder and Moens, 2010; Paetzold and Specia, 2015; Bingel and Søgaard, 2016). Such a statement may describe the essence of simplification as a research task, but it hides the fact that it is not always easy to decide what is easy for a particular user. This paper discusses why we need custom-tailored simplifications for individual users, and argues that previous research on non-adaptive text simplification has been too generic to unfold the full potential of text simplification.

Even when limiting ourselves to lexical substitution, i.e. the task of reducing the complexity of a document by replacing difficult words with easier-to-read synonyms, we see plenty of evidence that, for instance, dyslexics are highly individual in what material is deemed easy and complex (Ziegler et al., 2008). Lexi, which we introduce in this paper, is a free, open-source and easily extensible tool for adaptively learning what items specific users find difficult, using this information to provide better (lexical) simplification. Our system initially serves Danish, but is easily extended to further languages. For surveys of text simplification, including resources across languages, see Siddharthan (2014), Shardlow (2014a) and Collins-Thompson (2014).

6.1.1 *There is no one-size-fits-all solution to text simplification*

Text simplification is a diverse task, or perhaps rather a family of tasks, with a number of different target audiences that different papers and research projects have focused on. Among the most prominent target audiences are foreign language learners, for whom various approaches to simplifying text have been pursued, often focusing on lexical (Tweissi, 1998) but also sentence-level simplification (Liu and Matsumoto, 2016). Other notable groups that have been specifically targeted in text simplification research include dyslexics (Rello et al., 2013b), and the aphasic (Carroll et al., 1998), for whom particularly long words and sentences, but also certain surface forms such as specific character combinations, may pose difficulties. People on the autism spectrum have also been addressed, with the focus lying on reducing the amount of figurative expressions in a text or reducing syntactic complexity (Evans et al., 2014). Reading beginners (both children and adults) are another group with very particular needs, and text simplification research has tried to provide this group with methods to reduce the amount of high-register language and non-frequent words (De Belder and Moens, 2010).

Evidently, each target group has its own simplification needs, and there is considerable variation as to how well the specifics of what makes a text difficult is defined for each group and simplification strategy. While difficult items in a text may be identified more easily and generally for problems such as resolving pronoun reference, questions such as what makes a French word difficult for a native speaker of Japanese, or what dyslexic children consider a difficult character combination or an overly long sentence, are much harder to answer. Nevertheless, there is a vast body of work (Yatskar et al., 2010; Biran et al., 2011; Horn et al., 2014) that ventures to build very general-purpose simplification models from simplification corpora such as the Simple English Wikipedia corpus (Coster and Kauchak, 2011c), which has been edited by amateurs without explicit regard to a specific audience, and with rather vague guidelines as to what constitutes difficult or simple language.

Other work in simplification attempts to answer the above questions by inducing models from specifically compiled datasets, which for instance may have been collected by surveying specific target groups and asking them to indicate difficult material in a text. Yet even those approaches often cannot live up to the real challenges in simplification, seeing that we find very heterogeneous simplification needs also within target groups. Foreign language learners with different linguistic backgrounds (pertaining both to their native and second languages) will find very different aspects of the same foreign language difficult. Young readers in different school grades will quickly advance their reading habits and skills, and also within the same class or age reading levels may differ greatly. Likewise, people with autism exhibit very different manifestations of the type and degree of their condition (Alexander et al., 2016), also with respect to reading (Evans et al., 2014), just as there exist many different forms of cognitive impairments affecting literacy, including many different forms of dyslexia (Watson and Goldgar, 1988; Bakker, 1992; Ziegler et al., 2008). In fact, while there is a relatively strong agreement on the existence of some typologies of dyslexia or autism, specific typologies that have been proposed are heavily debated, such that it would not even be straightforward to create simplification tools for specific subtypes of these conditions.

From this it becomes apparent that in order to build simplification systems that truly help specific individuals, those systems have to be personalized or personalizable. Further, due to the frequent lack of insight into what an individual's specific reading problems are (and because any introspection is difficult to verify), such systems need to be able to learn themselves what those individual challenges are, and ultimately adapt to those.

6.1.2 *Obtaining individual data*

In order to learn specific reading challenges for an individual person, a simplification system needs individual data for this person, from which a personalized model can then be induced. This brings up the question of how best to obtain such data. A straightforward approach would be to ask each individual to provide ratings for some number of stimuli as they start using a simplification system. However, this would pose a relatively unnatural reading scenario, which might introduce a certain bias in the data and thus distort the induced model. Further, it might create a dissatisfying user experience, and users might not be willing to invest much time into such a calibration phase, especially when they perceive reading as a particularly strenuous activity. Yet perhaps most importantly, the model will not necessarily be well-adapted to the specific domains and genres that a specific user typically consumes text from.

As an alternative, we propose to collect data as the system is used, and to continuously update the system with feedback it collects from the user. In this way, the system can base its model on exactly those text types the user consumes. We discuss how feedback can be incorporated into a system in Section 6.3 and provide details on how this is implemented in our proposed system in Section 6.4.

6.1.3 *Contributions*

We present Lexi, an open source and easily extensible tool for adaptive, personalized text simplification. Lexi is based on an adaptive framework for lexical simplification that we also describe in this paper. This framework incorporates feedback from users, updating personalized simplification models such as to meet their individual simplification needs. Lexi is made publicly available under a CC-BY-NC license[24] at https://www.readwithlexi.net.

6.2 RELATED WORK

Perhaps the earliest contribution that focuses on on-demand lexical simplification is the work of Devlin and Unthank (2006), who present HAPPI, a web platform that allows users to request simplified versions of words, as well as other "memory jogging" pieces of information, such as related images.

Another example is the work of Azab et al. (2015), who present a web platform that allows users to select words they do not comprehend, then presents them with synonyms in order to facilitate comprehension. Notice that their approach does not simplify the selected complex words directly, it simply shows semantically equivalent alternatives that could be within the vocabulary known by the user.

The recent work of Paetzold and Specia (2016a) describes Anita, yet another web platform of this kind. It allows users to select complex words and then request a simplified version, related images, synonyms, definitions and translations. Paetzold and Specia (2016a) claim that their approach outputs customized simplifications depending on the user's profile, and evolves as users provide feedback on the output produced. However, they provide no details of the approach they use to do so, nor do they present any results showcasing its effectiveness.

Therefore not counting Paetzold and Specia (2016a) as work in *personalized* simplification, we are not aware of any previous approaches that address this. We further refer to related work on specific aspects of text simplification as they become relevant in the course of this paper.

6.3 ADAPTIVE TEXT SIMPLIFICATION

As we mapped out in the introduction, we devise a simplification system that continuously learns from user feedback and adapts to the user's simplification needs. This section discusses how such feedback can be incorporated into a *lexical simplification* model via online learning, and where in the lexical simplification pipeline it is sensible to implement adaptivity.

[24] https://creativecommons.org/licenses/by-nc/4.0/

6.3.1 *Adaptivity in the lexical simplification pipeline*

Lexical simplification, i.e. replacing single words with simpler synonyms, classically employs a pipeline approach illustrated in Figure 6.1 (Shardlow, 2014b; Paetzold and Specia, 2015). This pipeline consists of a four-step process, the first step of which is to identify simplification targets, i.e. words that the model believes will pose a difficulty for the user. This step is called *Complex Word Identification* (CWI) and has received a great deal of attention in the community, including two shared tasks (Paetzold and Specia, 2016b; Yimam et al., 2018). In a second step, known as *Substitution Generation*, synonyms are retrieved as candidate replacements for the target These are then filtered to match the context, resolving word sense ambiguities or stylistic mismatches, in *Substitution Selection*. Finally, those filtered candidate are ranked in order of simplicity in what is known as *Substitution Ranking* (SR).

Out of these four steps, we consider CWI and SR as the most natural ones to make adaptive, whereas generation and selecting candidates can be regarded as relatively independent from a specific user. In order to implement adaptivity, we propose to make use of online learning methods as discussed below and, departing from a seed model, train and maintain *user-specific models* as we collect feedback.

6.3.1.1 *Adaptive CWI*

Complex Word Identification is usually approached as a binary classification task, where the goal is to decide for some word *in context* whether or not it poses a difficulty to a reader. Existing datasets, for instance the ones used at previous CWI shared tasks (Paetzold and Specia, 2016b; Yimam et al., 2018), therefore provide a sentence and a target word (or multi-word expression) together with a binary label.

A model trained on this data with a learning algorithm based on gradient descent on *t* examples can now easily integrate newly collected data points into its parameters θ using an update rule such as

$$\theta^{(t+1)} = \theta^{(t)} - \eta \nabla_{\theta^{(t)}} J(\theta^{(t)}; x, y), \tag{6.1}$$

where x is a representation of a target word in context and y is a binary complexity label we receive from user feedback. As an alternative to gradient descent based algorithms, we can use other online learning models, e.g. the Perceptron algorithm. CWI datasets are typically not very large (between 2,500 and 5,500 positive examples per dataset in the mentioned shared tasks), such that data points sampled from users can quickly have an impact on a generic base model.[25]

[25] An alternative to traditional, one-size-fits-all approaches has recently been proposed by Bingel et al. (2018b), who use eye-tracking measures to induce personalized models to predict misreadings in children with reading difficulties.

6.3.1.2 *Adaptive Substitution Ranking*

Substitution Ranking has received relatively little attention in the community compared to CWI. Most lexical simplifiers rank candidates using unsupervised approaches. The earliest example is the approach of Carroll et al. (1998), who rank candidates according to their Kucera-Francis coefficients, which are calculated based on frequencies extracted from the Brown corpus (Rudell, 1993). Other unsupervised approaches, such as those of Ligozat et al. (2012) and Glavaš and Štajner (2015), go a step further and use metrics that incorporate multiple aspects of word complexity, including context-aware features such as n-gram frequencies and language model probabilities. But even though unsupervised rankers perform well in the task, they are incapable of learning from data, which makes them unsuitable for adaptive SR.

Our approach to adaptive SR is similar to our approach to adaptive CWI, namely to train an initial model over manually produced simplicity rankings, then continuously update them with new knowledge as Lexi users provide feedback on the simplifications they receive. The feedback in this scenario is composed of a complex word in context, a simplification produced by Lexi, and a binary rank provided by the user determining which word (complex or simplification) makes the sentence easier to understand. For that purpose, we need a supervised model that (i) supports online learning so that it can be efficiently updated after each session, and (ii) can learn from binary ranks.

Paetzold and Specia (2017b) offer some intuition on how this can be done. They exploit the fact that one can decompose a sequence of elements $\{e_1, e_2, ..., e_n\}$ with ranks $\{r_1, r_2, ..., r_n\}$ into a matrix $m \in \mathbb{R}^{n \times n}$, such that $m(i, j) = f(r_i, r_j)$, and function $f(r_i, r_j)$ estimates a value that describes the relationship between the ranks of elements e_i and e_j. For example, f could be described as:

$$f(r_i, r_j) = \begin{cases} 1 \text{ if } r_i < r_j \\ -1 \text{ if } r_i > r_j \\ 0 \text{ otherwise} \end{cases} \tag{6.2}$$

The ranker of Paetzold and Specia (2017b) uses a deep multi-layer perceptron that predicts each value of m individually. It takes as input feature representations of e_i and e_j, and produces a function f similar to the one depicted in Equation 6.2. Their approach would be perfectly capable of learning from the feedback produced by Lexi users, but it would be very difficult to train it through online learning, given that deep multi-layer perceptrons are characterized by a large number of parameters that are costly to optimize in an on-demand basis. We instead propose to employ an online learning model that has fewer parameters, e.g. logistic regression.

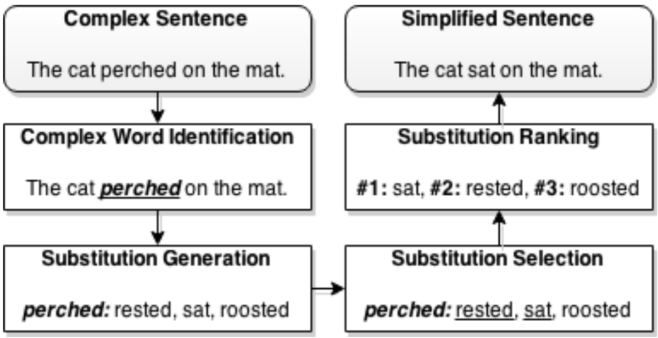

Complex Sentence
The cat perched on the mat.
Complex Word Identification
The cat ***perched*** on the mat.
Substitution Generation
perched: rested, sat, roosted
Substitution Selection
perched: rested, sat, roosted
Substitution Ranking
#1: sat, **#2:** rested, **#3:** roosted
Simplified Sentence
The cat sat on the mat.

Figure 6.1: Lexical simplification pipeline as identified by Shardlow (2014b). The simplification workflow consists of identifying simplification targets, i.e. words that pose a challenge to the reader. In the generation step, possible alternatives for each target are retrieved, which are then filtered in the selection step, eliminating words that do not fit the context. In the ranking step, the system finally orders the candidates by simplicity. Picture taken from Paetzold and Specia (2015).

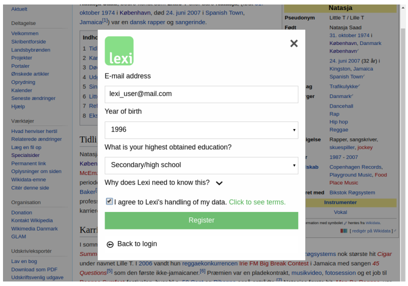
6.4 IMPLEMENTATION

Lexi consists of a client-side frontend and a server-side backend that communicate with each other via a RESTful API (Fielding, 2000), exchanging requests and responses as described further in 6.4.3. The client-server architecture allows for easy portability of the software to users, minimizing user-side installation efforts, hardware usage and dependencies on other libraries. It also centralizes the simplification engine, such that amendments to and maintenance of the latter need only be implemented on the server side.

Lexi is currently limited to performing lexical simplification. Note, however, that this is merely a limitation of the backend system, which only implements a lexical simplification system for now. From the frontend perspective, however, there are no limitations as to the nature and length of the simplified items in a text, and extending Lexi to support higher-level modes of simplification simply amounts to implementing a backend system supporting this.[26]

We initially focus on lexical simplification for a number of reasons: (i) We have existing baseline models that we expect to work well in a real-world setting. (ii) Given a relatively small number of parameters in those models, we expect fast adaptation to individual users from relatively little feedback. (iii) Compared to other forms of simplification, lexical simplification needs to make a selection from a relatively limited search space that is still reasonably diverse, such that we expect personalized models to make a difference more easily.

[26] Note that, in general, this paper describes the Lexi frontend and backend versions 1.0. Both parts of Lexi are under ongoing development, with details pertaining to the implementation possibly subject to change.



lexi

E-mail address

lexi_user@mail.com

Year of birth

1996

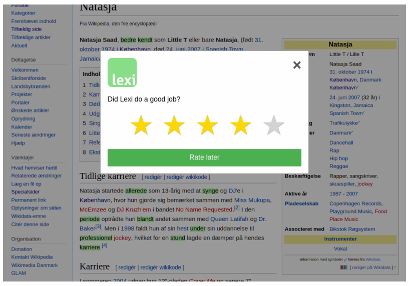
What is your highest obtained education?

Secondary/high school

Why does Lexi need to know this?

☑ I agree to Lexi's handling of my data. Click to see terms.

Figure 6.2: User registration form

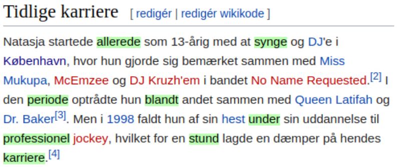


lexi

Did Lexi do a good job?

★★★★☆

Figure 6.3: Five-point rating form



Tidlige karriere [redigér | redigér wikikode]

Natasja startede allerede som 13-årig med at synge og DJ'e i København, hvor hun gjorde sig bemærket sammen med Miss Mukupa, McEmzee og DJ Kruzh'em i bandet No Name Requested.[2] I den periode optrådte hun blandt andet sammen med Queen Latifah og Dr. Baker[3]. Men i 1998 faldt hun af sin hest under sin uddannelse til professionel jockey, hvilket for en stund lagde en dæmper på hendes karriere.[4]

Figure 6.4: Simplification spans are marked up in light green. As the user clicks on a simplification span, the currently displayed word is replaced with an alternative.

6.4.1 *Frontend*

Lexi's frontend is implemented in JavaScript and jQuery under the Mozilla WebExtension framework, supported by most modern browsers.[27] WebExtensions employ *content scripts* to modify a webpage upon certain specified events, for instance a click on some page element. The remainder of this section describes Lexi's basic usage as the user registers an account and asks the system for simplifications, thereby illustrating the user interface and sketching the inner workings of the frontend.

6.4.1.1 *User log-in and registration*

Upon installation of the Lexi extension in the browser, the user is prompted to register an account, providing an email address as well as basic demographic information (year of birth and educational level, see Figure 6.2). This information is sent to the backend using its registration endpoint (see Table 6.1). If the user has previously created an account and simply reinstalled the extension, they may also just provide their email address to keep using their existing profile. The user's email address is stored locally in the browser, where it is kept until the browser storage is cleared or the extension is uninstalled.

[27] https://developer.mozilla.org/en-US/Add-ons/WebExtensions

6.4.1.2 *Simplification requests and display*

Whenever the user visits a webpage, the extension injects an event listener into the page, which triggers upon the selection of some text and offers the user to simplify the selected content in the form of a small button that is displayed just above the selection. When this button is clicked, the extension retrieves the user's email address from the browser storage (prompting the user to log in if no email address is stored) and verifies that a user with that email address exists in the backend's database, using the login endpoint as given in Table 6.1. The script then submits a *simplification request* to the backend's simplification endpoint, enclosing a JSON object that contains the user's email address (used by the backend to retrieve the personal simplification model) and the HTML code of the element containing the text selection. See Appendix A.1.1 for an example.

The response from the backend then transmits a JSON object with augmented HTML, where <`span`> elements with unique IDs are wrapped around simplification targets. The response object further contains an array of *simplification objects*, each of which in turn contains a list of synonyms ordered by simplicity ranking (including the target). An example is given in Appendix A.1.2. The content script replaces the original source with the augmented HTML and displays each simplification span with a light green background color (see Figure 6.4). The script then shifts through the simplification alternatives for a given target whenever the user clicks on the respective span on the page, advancing one alternative per click and reverting to the first alternative at the end of list. The original item is marked in a slightly but discernibly darker shade than the proposed simplifications.

6.4.1.3 *User feedback*

In order to provide personalized simplifications and to adapt to individual users, Lexi needs to be able to decide which alternative a user prefers over the others for every target. In a classical, controlled annotation setting, one would probably present subjects with a set of alternatives and have them rank these or pick a single favorite. However, as Lexi aims to provide as natural and smooth a reading scenario as possible to its users, explicitly asking for such feedback would critically obstruct the reading process.

Lexi therefore interprets whatever final selection a user makes for some simplification span as their preferred alternative in this context.[28] As the user finally navigates away from the webpage that Lexi was invoked on, Lexi solicits feedback from the user on a five-point

[28] In the instructions, users are made aware of this. The frontend further keeps track of how many times the user clicked on a given simplification span, thus providing the backend with information such as how many times the user clicked through the entire list, or whether perhaps no alternatives were solicited for some item.

scale (see Figure 6.3) and submits this rating along with the simplification objects and their final selections (and click-through counts) to the feedback endpoint of the backend.[29] See Appendix A.1.3 for an example of the feedback.

6.4.1.4 *Qualitative evaluation of usability*

The frontend design was developed in close collaboration with Nota, the Danish Library and Expertise Center for people with reading disabilities.[30] In February 2018, the software was intensively tested by four dyslexic members of Nota, all female students in secondary education and aged between 20 and 30. Each test started with a short preliminary interview in which the subjects were asked about their age, occupation/study field, reading habits, degree of dyslexia and use of browser extensions. The subjects were then given the possibility to watch an introduction video (of 1:30 min length) outlining Lexi's basic functionality and user interface. Two of the four subjects opted for this, while the other two decided to skip the video as they do not usually watch introduction videos when using new software. Next, the subjects were asked to locate Lexi in the Chrome Webshop, install it in the browser and create a user account. Once set up, each subject navigated to a site of her choice and used Lexi to receive simplifications as outlined in 6.4.1.2. The two subjects who had not watched the video did so now, and both declared they gained further insight into Lexi's functionality through the video, but that it was not crucial in order to understand its basic usage.

In qualitative interviews directly succeeding each test, the test subjects overall reacted very positively to the prospect of a personalized simplification tool in general, and to Lexi and its design in particular.[31] The test subjects suggested a number of improvements, most of which have now been implemented. One suggested improvement, which we have not been able to implement but intend to do so for a future version, is the support for multilingual simplification. Two subjects said they would greatly appreciate this, as much of their study material is only available in English.

[29] More correctly, feedback is not solicited when the user actual navigates away from the page, as security restrictions in browsers disallow custom scripts to run upon closing a page. Instead, Lexi asks for feedback via a small notification box in the upper right corner of the page, which pops up as the operating system's *focus* changes to a different window, or when the mouse leaves the browser's viewport (e.g. for the address bar).

[30] `http://www.nota.dk`

[31] An informal evaluation of the software on a 5-point scale (with 1 being worst and 5 best) yielded two ratings of 5, one 4 and one 3.

6.4.2 *Backend*

Lexi's backend consists of a simplification system, implemented in Python 3.5, and a database that stores user information and their simplification histories.

6.4.2.1 *Simplification system*

As stated above, Lexi's simplification system currently focuses on lexical simplification, abiding to the *de-facto* standard pipeline depicted in Figure 6.1. Since Lexi lets users choose which words they wish to have simplified, it does not employ any automatic CWI.[32] Below we sketch Lexi's simplification system as it receives simplification requests from the frontend. As our lexical simplification approach is sensitive to the context of a word, Lexi's first step is to preprocess the HTML source transmitted from the frontend, identifying the boundaries of the sentence that contains the target word, if any.[33]

For Substitution Generation, Lexi's backend implements the embeddings-based approach inspired by the contributions of Glavaš and Štajner (2015) and Paetzold and Specia (2016d). In their work, they extract as candidate substitutions the N words with the highest cosine similarity with a target word. As Danish, the language currently served by Lexi, is not as well-resourced as for example English, Lexi extends the embedding-based Substitution Generation approach by using an ensemble of embeddings models that are trained independently on different text sources, the Danish Wikipedia and a news corpus.[34] The overall similarity score for a target-candidate pair is then defined as the mean score across these embeddings models. Lexi returns the ten most similar candidates whose mean similarity score exceeds some configurable threshold. Alternatively, Lexi allows to generate synonyms from a simple dictionary, in the case of Danish using the Danish WordNet (Pedersen et al., 2009), yet this approach suffers from severely reduced coverage compared to word embeddings.

Once generated, the candidates are filtered during Substitution Selection by an unsupervised boundary ranker (Paetzold and Specia, 2016d). In this approach, a supervised ranker is trained with instances gathered in an unsupervised fashion: we generate candidate substitutions for complex words using our generation approach, then assign label 1 to the complex words and 0 to the generated candidates. The boundary between the two classes is then used to rank and filter

[32] We do plan, however, to implement CWI as the user solicits simplifications for longer text passages or entire pages.

[33] In order to reduce bandwidth and modify the page more easily, the frontend only transmits the HTML source of the least HTML node fully containing the selection, which typically is a paragraph (`<p>`), but may also be a single word contained in a heading (e.g. `<h1>`), in which case no context is available. Sentence boundaries are identified using NLTK.

[34] https://ordnet.dk/korpusdk

candidates. Paetzold and Specia (2016d) show that this is a state-of-the-art approach that outperforms all earlier supervised and unsupervised strategies. Given a target word and a set of generated candidate substitutions, the model ranks the candidates based on how far in the positive side of the data they are, then selects 65% of the highest ranking ones.

Finally, the selected candidates are ranked with a supervised Substitution Ranking model following the approach we outlined in Section 6.3.1.2. It is during this step that Lexi is capable of producing customized output based on the user's needs, and to evolve based on the user's feedback. Lexi employs a pairwise online logistic regression model that learns to quantify the simplicity difference between two candidate substitutions. Given an unseen set of candidate substitutions, the regressor estimates the simplicity difference between each candidate pair, then ranks all candidates based on their average score.

Note that the user's feedback, sent by the frontend, consists of a set S and an index i, where S is the full set of suggested synonyms, including the target, and i is the index of the item in S that the user finally selected. As the regressor, however, learns from pairwise rankings, Lexi passes all pairs $\{\langle S_i, S_j \rangle | j \neq i\}$ to the regressor, i.e. it pairs the selected item with all others and updates the ranker accordingly, postulating that the selected item is easier for this user than each other suggestion.

Using a seed dataset of complex-simple word correspondences in context, we train a default model that produces initial simplifications as a user solicits simplifications for the first time.[35] As Lexi receives feedback for this user for the first time, the seed model is copied and personalized with the first batch of feedback, then this model is saved for later requests by this user.

6.4.2.2 *Database*

Lexi stores user information and simplification histories in a PostgreSQL database. More specifically, it employs three different tables, called `users`, `models` and `sessions`. In the first of these, it links a unique, numerical user ID to a user email address, and stores when the user first and last used Lexi. It further contains the demographic information the user provides at registration, i.e. their year of birth and educational status. The `models` table stores a path to the serialized personal model for each user ID. Finally, the `sessions` table stores each simplification request issued to the backend with a unique

[35] Such a seed dataset is not necessarily available for any language. However, in its absence, a seed model could either be trained with simple heuristics, e.g. replacing infrequent words with higher-frequency synonyms. Alternatively, the system could choose to initially rank candidates with such a heuristic and only start learning once the first feedback is available.

URI path	Input	Returns
`/simplify`	User ID; page HTML	Augmented HTML, simplification objects
`/login`	User email address	If successful: User ID, else error code
`/register`	User information	If successful: User ID, else error code
`/feedback`	User ID; simplification objects updated with selections; rating	Status code (successful update or error)

Table 6.1: RESTful API endpoints defined by Lexi's backend.

session ID, the respective user ID, a time stamp for the session start and one for the submission of feedback, the webpage URL, simplification objects serialized as JSON, the provided rating and finally the frontend version number used in this session.

6.4.3 *Communication between backend and frontend*

Lexi's backend offers a RESTful API implemented in Python 3.5, using the Flask package.[36] The services available through HTTP POST requests, with their URI paths listed in Table 6.1. Input and output values are communicated via a JSON-based protocol exemplified in the appendix. Lexi further defines a set of error codes for easier troubleshooting and flexible internationalization of the frontend via the i18n API used by WebExtensions.

6.4.4 *Language support and extensibility*

Lexi's design does not impose any restrictions on the support of new (written) languages, including right-to-left or non-alphabetic writing systems. In fact, supporting a new language simply amounts to providing a new language-specific simplification pipeline as illustrated in Figure 6.1.

Depending on the specific implementation of the simplification system, certain resources are however needed to induce a first seed model for simplification. Most centrally, this pertains to Substitution Generation, where a synonym database or good word embeddings are required in the case of lexical simplification, or a reliable paraphrase module in the case of higher-level simplification. With respect to Substitution Ranking, the availability of resources such as simpli-

[36] http://flask.pocoo.org/

fication corpora is less critical, as simple heuristics (e.g. simplicity proxies such as length and frequency) might give a reasonable baseline upon which the system can then improve through user feedback.

Lexi currently does not offer multilingual support, but is confined to one language per backend instance. Supporting multilingual simplification could be implemented through a language identification module upstream to the set of simplification pipelines, consisting of one pipeline per language. This raises the interesting question whether knowledge about one user's simplification preferences in one language could be transferred to another language. Support for this hypothesis comes, among others, from the cross-lingual track in the recent CWI shared task by Yimam et al. (2018).

6.4.5 *Ethical and legal considerations*

As any software interacting with users and storing information on them, Lexi is naturally subject to ethical and legal concerns, especially those regarding privacy. The EU General Data Protection Regulation (GDPR), for instance, defines a number of regulations such as the clear statement of terms and conditions or that users need to be provided, upon request, with full access to whatever data is stored on them. Lexi does not explicitly store users' names, but in many cases they will be encoded in email addresses. Personally identifiable information may also be stored in the form of simplified text that is logged in the database, for instance if Lexi is used on a user's personal social media profile. The above also highlights the need for encrypted communication between the client and the server, which is safeguarded through TLS encryption over the HTTPS protocol.

Ethical concerns pertaining to text simplification arise when infelicitous simplifications distort the meaning of a text and thus potentially misinform the reader. This is difficult to completely rule out, such that the user should clearly be informed of this possibility. Other concerns revolve around the hypothesis that reducing text complexity will "dumb down" the material and keep users at a low reading level by under-challenging them (Long and Ross, 1993). However, as Rello et al. (2013b) point out, "anything which might help [dyslexics] to subjectively perceive reading as being easier, can potentially help them to avoid this vicious circle [of reading less and staying on a low reading level], even if no significant improvement in readability can be demonstrated."

6.5 AVAILABILITY AND APPLICATIONS

The Lexi software and code, including its backend and frontend, are freely available for non-commercial use under a CC-BY-NC license, obtainable at `https://www.readwithlexi.net`. Researchers can set up their own, customized version of the software and distribute the browser extension to users. It is straightforward to modify features of the software such as offered languages or the exact resources used to induce the initial models.

Besides its core functionality, which we mapped out in the previous sections, Lexi has a number of alternate use cases, which we discuss in this section.

PRELOADED SIMPLIFICATIONS Lexi's primary use case, as described earlier, is to provide simplifications to users as they select a span of text, which circumvents the need for a CWI module as only such items are simplified that the user explicitly solicits replacements for. Alternatively, users may wish to have the entire page simplified before they start reading. Lexi currently implements this functionality, letting the user solicit simplifications for the entire site via a click on the Lexi icon. As there is no personalized CWI module implemented yet, simplification targets are identified via a confidence threshold during Substitution Generation.

EVALUATION OF SIMPLIFICATION QUALITY Via its rating function (Figure 6.3), Lexi continuously tracks user satisfaction as a means of evaluating synchronic simplification quality as well as the diachronic development of model adaptation. An adaptive model that is continuously customized is expected to gradually improve the average rating it receives from the user.

DATA COLLECTION Lexi makes it possible to collect user choices over a longer period in order to create bigger simplification datasets. If sufficiently homogeneous subgroups can be identified across users, this data may give insight into their simplification needs, to build better simplification models for them.

Other plausible approaches may understand different users as different *tasks* and apply multi-task learning methods to transfer knowledge between users, thus both regularizing the models for the individual user and increasing the available amount of data that the individual models can be learned from.

6.6 CONCLUSION AND FUTURE WORK

This paper is a first work in personalized, adaptive text simplification, a direction of research motivated by the observation that generic, user-independent simplification systems cannot fully unfold their potential in making text simpler for specific end users. We propose a framework for adaptive lexical simplification, outlining how user feedback can be used to gradually enhance and personalize text simplification. As a concrete first solution to the problem, we present

Lexi, an open-source tool for personalized, adaptive text simplification that has been very positively evaluated in a first usability test. In its current implementation, Lexi focuses on lexical simplification in Danish. An extension to other languages is simple, requiring only a medium-sized monolingual corpus on which a language model and word embeddings can be trained.

In future work, we aim to extend the proposed framework to sentence-level simplifications. We further plan to implement support for multilingual simplification.

Part IV

MULTI-TASK LEARNING FOR TEXT SIMPLIFICATION

7

IDENTIFYING BENEFICIAL TASK RELATIONS FOR MULTI-TASK LEARNING IN DEEP NEURAL NETWORKS

ABSTRACT

Multi-task learning (MTL) in deep neural networks for NLP has recently received increasing interest due to some compelling benefits, including its potential to efficiently regularize models and to reduce the need for labeled data. While it has brought significant improvements in a number of NLP tasks, mixed results have been reported, and little is known about the conditions under which MTL leads to gains in NLP. This paper sheds light on the specific task relations that can lead to gains from MTL models over single-task setups.

7.1 INTRODUCTION

Multi-task learning is receiving increasing interest in both academia and industry, with the potential to reduce the need for labeled data, and to enable the induction of more robust models. The main driver has been empirical results pushing state of the art in various tasks, but preliminary theoretical findings guarantee that multi-task learning works under various conditions. Some approaches to multi-task learning are, for example, known to work when the tasks share optimal hypothesis classes (Baxter, 2000) or are drawn from related sample generating distributions (Ben-David and Borbely, 2008).

In NLP, multi-task learning typically involves very heterogeneous tasks. However, while great improvements have been reported (Luong et al., 2016; Klerke et al., 2016), results are also often mixed (Collobert and Weston, 2008; Søgaard and Goldberg, 2016; Martínez Alonso and Plank, 2017), and theoretical guarantees no longer apply. The question *what task relations guarantee gains or make gains likely in NLP* remains open.

CONTRIBUTIONS This paper presents a systematic study of *when* and *why* MTL works in the context of sequence labeling with deep recurrent neural networks. We follow previous work (Klerke et al., 2016; Søgaard and Goldberg, 2016; Bollman and Søgaard, 2016; Plank, 2016; Braud et al., 2016; Martínez Alonso and Plank, 2017) in studying the set-up where hyperparameters from the single task architectures are reused in the multi-task set-up (no additional tuning), which makes predicting gains feasible. Running MTL experiments on 90 task configurations and comparing their performance to single-task setups, we identify data characteristics and patterns in single-task learning that predict task synergies in deep neural networks. Both the LSTM code used for our single-task and multi-task models, as well as the script we used for the analysis of these, are available at github.com/jbingel/eacl2017_mtl.

7.2 RELATED WORK

In the context of structured prediction in NLP, there has been very little work on the conditions under which MTL works. Luong et al. (2016) suggest that it is important that the auxiliary data does not outsize the target data, while Benton et al. (2017) suggest that multi-task learning is particularly effective when we only have access to small amounts of target data. Martínez Alonso and Plank (2017) present a study on different task combinations with dedicated main and auxiliary tasks. Their findings suggest, among others, that success depends on how uniformly the auxiliary task labels are distributed.

Mou et al. (2016) investigate multi-task learning and its relation to transfer learning, and under which conditions these work between a set of sentence classification tasks. Their main finding with respect to multi-task learning is that success depends largely on "how similar in semantics the source and target datasets are", and that it generally bears close resemblance to transfer learning in the effect it has on model performance.

7.3 MULTI-TASK LEARNING

While there are many approaches to multi-task learning, hard parameter sharing in deep neural networks (Caruana, 1993) has become extremely popular in recent years. Its greatest advantages over other methods include (i) that it is known to be an efficient regularizer, theoretically (Baxter, 2000), as well as in practice (Søgaard and Goldberg, 2016); and (ii) that it is easy to implement.

The basic idea in hard parameter sharing in deep neural networks is that the different tasks share some of the hidden layers, such that these learn a joint representation for multiple tasks. Another conceptualization is to think of this as regularizing our target model by doing model interpolation with auxiliary models in a dynamic fashion.

Multi-task linear models have typically been presented as matrix regularizers. The parameters of each task-specific model makes up a row in a matrix, and multi-task learning is enforced by defining a joint regularization term over this matrix. One such approach would be to define the joint loss as the sum of losses and the sum of the singular values of the matrix. The most common approach is to regularize learning by the sum of the distances of the task-specific models to the

Task	Size	# Labels	Tok/typ	%OOV	$H(y)$	$\|\|X\|\|_F$	JSD	F_1
CCG	39,604	1,285	23.08	1.13	3.28	981.3	0.41	86.1
CHU	8,936	22	12.01	1.35	1.84	466.4	0.47	93.9
COM	9,600	2	9.47	0.99	0.47	519.3	0.44	51.9
FNT	3,711	2	8.44	1.79	0.51	286.8	0.30	58.0
POS	1,002	12	3.24	14.15	2.27	116.9	0.24	82.6
HYP	2,000	2	6.14	2.14	0.47	269.3	0.48	39.3
KEY	2,398	2	9.10	4.46	0.61	289.1	0.39	64.5
MWE	3,312	3	9.07	0.73	0.53	217.3	0.18	43.3
SEM	15,465	73	11.16	4.72	2.19	614.6	0.35	70.8
STR	3,312	118	9.07	0.73	2.43	217.3	0.26	61.5

Table 7.1: Dataset characteristics for the individual tasks as defined in Table 7.2, as well as single-task model performance on test data (micro-averaged F_1).

model mean. This is called mean-constrained learning. Hard parameter sharing can be seen as a very crude form of mean-constrained learning, in which parts of all models (typically the hidden layers) are enforced to be identical to the mean.

Since we are only forcing parts of the models to be identical, each task-specific model is still left with wiggle room to model heterogeneous tasks, but the expressivity is very limited, as evidenced by the inability of such networks to fit random noise (Søgaard and Goldberg, 2016).

7.3.1 *Models*

Recent work on multi-task learning of NLP models has focused on sequence labeling with recurrent neural networks (Klerke et al., 2016; Søgaard and Goldberg, 2016; Bollman and Søgaard, 2016; Plank, 2016; Braud et al., 2016; Martínez Alonso and Plank, 2017), although sequence-to-sequence models have been shown to profit from MTL as well (Luong et al., 2016). Our multi-task learning architecture is similar to the former, with a bi-directional LSTM as a single hidden layer of 100 dimensions that is shared across all tasks. The inputs to this hidden layer are 100-dimensional word vectors that are initialized with pretrained GloVe embeddings, but updated during training. The embedding parameters are also shared. The model then generates predictions from the bi-LSTM through task-specific dense projections. Our model is symmetric in the sense that it does not distinguish between main and auxiliary tasks.

In our MTL setup, a training step consists of uniformly drawing a training task, then sampling a random batch of 32 examples from the task's training data. Every training step thus works on exactly one task, and optimizes the task-specific projection and the shared parameters using Adadelta. As already mentioned, we keep hyperparameters fixed across single-task and multi-task settings, making our results only applicable to the scenario where one wants to know whether MTL works in the current parameter setting (Collobert and Weston, 2008; Klerke et al., 2016; Søgaard and Goldberg, 2016; Bollman and Søgaard, 2016; Plank, 2016; Braud et al., 2016; Martínez Alonso and Plank, 2017).

7.3.2 *Tasks*

In our experiments below, we consider the following ten NLP tasks, with one dataset for each task. Characteristics of the datasets that we use are summarized in Table 7.1.

1. **CCG Tagging** (CCG) is a sequence tagging problem that assigns a logical type to every token. We use the standard splits for CCG super-tagging from the CCGBank (Hockenmaier and Steedman, 2007).

2. **Chunking** (CHU) identifies continuous spans of tokens that form syntactic units such as noun phrases or verb phrases. We use the standard splits for syntactic chunking from the English Penn Treebank (Marcus et al., 1993).

3. **Sentence Compression** (COM) We use the publicly available subset of the Google Compression dataset (Filippova and Altun, 2013), which has token-level annotations of word deletions.

4. **Semantic frames** (FNT) We use FrameNet 1.5 for jointly predicting target words that trigger frames, and deciding on the correct frame in context.

5. **POS tagging** (POS) We use a dataset of tweets annotated for Universal part-of-speech tags (Petrov et al., 2011).

6. **Hyperlink Prediction** (HYP) We use the hypertext corpus from Spitkovsky et al. (2010) and predict what sequences of words have been bracketed with hyperlinks.

7. **Keyphrase Detection** (KEY) This task amounts to detecting keyphrases in scientific publications. We use the SemEval 2017 Task 10 dataset.

8. **MWE Detection** (MWE) We use the Streusle corpus (Schneider and Smith, 2015) to learn to identify multi-word expressions (*on my own, cope with*).

Data features	
Size	Number of training sentences.
# Labels	The number of labels.
Tokens/types	Type/token ratio in training data.
OOV rate	Percentage of training words not in GloVe vectors.
Label Entropy	Entropy of the label distribution.
Frobenius norm	$\|\|X\|\|_F = [\sum_{i,j} X_{i,j}^2]^{1/2}$, where $X_{i,j}$ is the frequency of term j in sentence i.
JSD	Jensen-Shannon Divergence between train and test bags-of-words.
Learning curve features	
Curve gradients	See text.
Fitted log-curve	See text.

Table 7.2: Task features

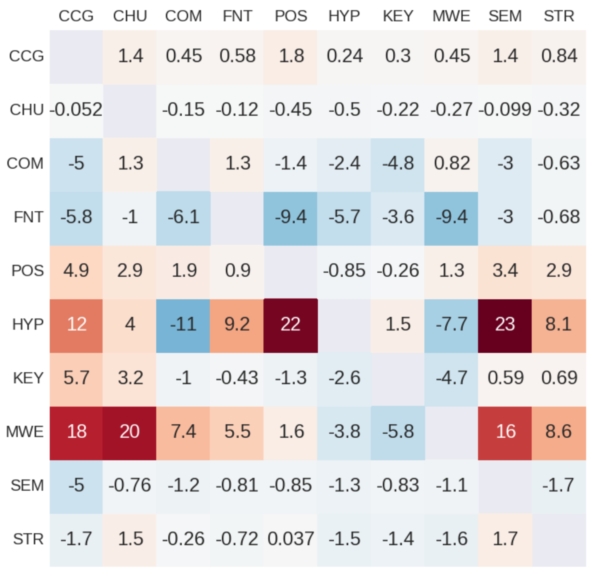
9. **Super-sense tagging** (sem) We use the standard splits for the Semcor dataset, predicting coarse-grained semantic types of nouns and verbs (super-senses).

10. **Super-sense Tagging** (str) As for the MWE task, we use the Streusle corpus, jointly predicting brackets and coarse-grained semantic types of the multi-word expressions.

7.4 experiments

We train single-task bi-LSTMs for each of the ten tasks, as well as one multi-task model for each of the pairs between the tasks, yielding 90 directed pairs of the form $\langle \mathcal{T}_{main}, \{\mathcal{T}_{main}, \mathcal{T}_{aux}\}\rangle$. The single-task models are trained for 25,000 batches, while multi-task models are trained for 50,000 batches to account for the uniform drawing of the two tasks at every iteration in the multi-task setup. The relative gains and losses from MTL over the single-task models (see Table 7.1) are presented in Figure 7.1, showing improvements in 40 out of 90 cases. We see that chunking and high-level semantic tagging generally contribute most to other tasks, while hyperlinks do not significantly improve any other task. On the receiving end, we see that multiword and hyperlink detection seem to profit most from several auxiliary tasks. Symbiotic relationships are formed, e.g., by POS and CCG-tagging, or MWE and compression.

We now investigate whether we can predict gains from MTL given features of the tasks and single-task learning characteristics. We will use the induced meta-learning for analyzing what such characteristics are predictive of gains.



	CCG	CHU	COM	FNT	POS	HYP	KEY	MWE	SEM	STR
CCG		1.4	0.45	0.58	1.8	0.24	0.3	0.45	1.4	0.84
CHU	-0.052		-0.15	-0.12	-0.45	-0.5	-0.22	-0.27	-0.099	-0.32
COM	-5	1.3		1.3	-1.4	-2.4	-4.8	0.82	-3	-0.63
FNT	-5.8	-1	-6.1		-9.4	-5.7	-3.6	-9.4	-3	-0.68
POS	4.9	2.9	1.9	0.9		-0.85	-0.26	1.3	3.4	2.9
HYP	12	4	-11	9.2	22		1.5	-7.7	23	8.1
KEY	5.7	3.2	-1	-0.43	-1.3	-2.6		-4.7	0.59	0.69
MWE	18	20	7.4	5.5	1.6	-3.8	-5.8		16	8.6
SEM	-5	-0.76	-1.2	-0.81	-0.85	-1.3	-0.83	-1.1		-1.7
STR	-1.7	1.5	-0.26	-0.72	0.037	-1.5	-1.4	-1.6	1.7	

Figure 7.1: Relative gains and losses (in percent) over main task micro-averaged F_1 when incorporating auxiliary tasks (columns) compared to single-task models for the main tasks (rows).

Specifically, for each task considered, we extract a number of dataset-inherent features (see Table 7.2) as well as features that we derive from the learning curve of the respective single-task model. For the curve gradients, we compute the gradients of the loss curve at 10, 20, 30, 50 and 70 percent of the 25,000 batches. For the fitted log-curve parameters, we fit a logarithmic function to the loss curve values, where the function is of the form: $L(i) = a \cdot \ln(c \cdot i + d) + b$. We include the fitted parameters a and c as features that describe the steepness of the learning curve. In total, both the main and the auxiliary task are described by 14 features. Since we also compute the main/auxiliary ratios of these values, each of our 90 data points is described by 42 features that we normalize to the $[0, 1]$ interval. We binarize the results presented in Figure 7.1 and use logistic regression to predict benefits or detriments of MTL setups based on the features computed above.[37]

7.4.1 *Results*

The mean performance of 100 runs of randomized five-fold cross-validation of our logistic regression model for different feature combinations is listed in Table 7.3. The first observation is that there is a strong signal in our meta-learning features. In almost four in five cases, we can predict the outcome of the MTL experiment from the data and the single task experiments, which gives validity to our

[37] An experiment in which we tried to predict the magnitude of the losses and gains with linear regression yielded inconclusive results.

	Acc.	F_1 (gain)
Majority baseline	0.555	0.615
All features	0.749	0.669
Best, data features only	0.665	0.542
Best combination	0.785	0.713

Table 7.3: Mean performance across 100 runs of 5-fold CV logistic regression.

feature analysis. We also see that the features derived from the single task inductions are the most important. In fact, using only data-inherent features, the F_1 score of the positive class is worse than the majority baseline.

7.4.2 *Analysis*

Table 7.4 lists the coefficients for all 42 features. We find that features describing the learning curves for the main and auxiliary tasks are the best predictors of MTL gains. The ratios of the learning curve features seem less predictive, and the gradients around 20-30% seem most important, after the area where the curve typically flattens a bit (around 10%). Interestingly, however, these gradients correlate in opposite ways for the main and auxiliary tasks. The pattern is that if the main tasks have flattening learning curves (small negative gradients) in the 20-30% percentile, but the auxiliary task curves are still relatively steep, MTL is more likely to work. In other words, *multi-task gains are more likely for target tasks that quickly plateau with non-plateauing auxiliary tasks*. We speculate the reason for this is that multi-task learning can help target tasks that get stuck early in local minima, especially if the auxiliary task does not always get stuck fast.

Other features that are predictive include the number of labels in the main task, as well as the label entropy of the auxiliary task. The latter supports the hypothesis put forward by Martínez Alonso and Plank (2017) (see Related work). Note, however, that this may be a side effect of tasks with more uniform label distributions being easier to learn. The out-of-vocabulary rate for the target task also was predictive, which makes sense as the embedding parameters are also updated when learning from the auxiliary data.

Less predictive features include Jensen-Shannon divergences, which is surprising, since multi-task learning is often treated as a transfer learning algorithm (Søgaard and Goldberg, 2016). It is also surprising to see that size differences between the datasets are not very predictive.

Feature	Task	Coefficient
Curve grad. (30%)	Main	-1.566
Curve grad. (20%)	Main	-1.164
Curve param. c	Main	1.007
# Labels	Main	0.828
Label Entropy	Aux	0.798
Curve grad. (30%)	Aux	0.791
Curve grad. (50%)	Main	0.781
OOV rate	Main	0.697
OOV rate	Main/Aux	0.678
Curve grad. (20%)	Aux	0.575
Fr. norm	Main	-0.516
# Labels	Main/Aux	0.504
Curve grad. (70%)	Main	0.434
Label entropy	Main/Aux	-0.411
Fr. norm	Aux	0.346
Tokens/types	Main	-0.297
Curve param. a	Aux	-0.297
Curve grad. (70%)	Aux	-0.279
Curve grad. (10%)	Aux	0.267
Tokens/types	Aux	0.254
Curve param. a	Main/Aux	-0.241
Size	Aux	0.237
Fr. norm	Main/Aux	-0.233
JSD	Aux	-0.207
# Labels	Aux	-0.184
Curve param. c	Aux	-0.174
Tokens/types	Main/Aux	-0.117
Curve param. c	Main/Aux	-0.104
Curve grad. (20%)	Main/Aux	0.104
Label entropy	Main	-0.102
Curve grad. (50%)	Aux	-0.099
Curve grad. (50%)	Main/Aux	0.076
OOV rate	Aux	0.061
Curve grad. (30%)	Main/Aux	-0.060
Size	Main	-0.032
Curve param. a	Main	0.027
Curve grad. (10%)	Main/Aux	0.023
JSD	Main	0.019
JSD	Main/Aux	-0.015
Curve grad. (10%)	Main	$6 \cdot 10^{-2}$
Size	Main/Aux	$-6 \cdot 10^{-3}$
Curve grad. (70%)	Main/Aux	$-4 \cdot 10^{-4}$

Table 7.4: Predictors of MTL benefit by logistic regression model coefficient (absolute value).

7.5 CONCLUSION AND FUTURE WORK

We present the first systematic study of when MTL works in the context of common NLP tasks, when single task parameter settings are also applied for multi-task learning. Key findings include that MTL gains are predictable from dataset characteristics and features extracted from the single-task inductions. We also show that the most predictive features relate to the single-task learning curves, suggesting that MTL, when successful, often helps target tasks out of local minima. We also observed that label entropy in the auxiliary task was also a good predictor, lending some support to the hypothesis in Martínez Alonso and Plank (2017); but there was little evidence that dataset balance is a reliable predictor, unlike what previous work has suggested.

In future work, we aim to extend our experiments to a setting where we optimize hyperparameters for the single- and multi-task models individually, which will give us a more reliable picture of the effect to be expected from multi-task learning in the wild. Generally, further conclusions could be drawn from settings where the joint models do not treat the two tasks as equals, but instead give more importance to the main task, for instance through a non-uniform drawing of the task considered at each training iteration, or through an adaptation of the learning rates. We are also interested in extending this work to additional NLP tasks, including tasks that go beyond sequence labeling such as language modeling or sequence-to-sequence problems.

ACKNOWLEDGMENTS

For valuable comments, we would like to thank Dirk Hovy, Yoav Goldberg, the attendants at the second author's invited talk at the Danish Society for Statistics, as well as the anonymous reviewers. This research was partially funded by the ERC Starting Grant LOWLANDS No. 313695, as well as by Trygfonden.

8

CROSS-LINGUAL COMPLEX WORD IDENTIFICATION WITH MULTITASK LEARNING

ABSTRACT

We approach the 2018 Shared Task on Complex Word Identification by leveraging a cross-lingual multitask learning approach. Our method is highly language agnostic, as evidenced by the ability of our system to generalize across languages, including languages for which we have no training data. In the shared task, this is the case for French, for which our system achieves the best performance. We further provide a qualitative and quantitative analysis of which words pose problems for our system.

8.1 INTRODUCTION

Complex word identification (CWI) is the task of predicting whether a certain word might be difficult for a reader to understand and is typically used as a first step in (lexical) simplification pipelines (Shardlow, 2014b; Paetzold and Specia, 2015, 2016b). This task has received significant attention from the community over the past few years, leading to two shared tasks and several other publications (Shardlow, 2013a,b).

This paper presents our submission to the CWI 2018 shared task (Yimam et al., 2018), at the 13th Workshop on Innovative Use of NLP for Building Educational Applications. This task includes tracks targeting four languages: English, Spanish, German and French. For each of these languages, the task involves prediction of binary labels of whether any of a range of annotators deemed some word or phrase complex, or prediction of the ratio of those who did. The task further differs from previous approaches to CWI in extending the definition of the target units from the word level to multi-word expressions, such that annotations in the training and test set spanned wider stretches of text than single tokens.

Another difference from previous approaches to CWI is that the data is annotated by a mixture of native and non-native speakers, posing an interesting challenge to reconcile the potentially different complexity assessments of these groups.

One challenge in the CWI 2018 shared task is the fact that one of the languages under consideration (French) does not have any training data available. We approach this problem by exploring a combination of multitask learning and cross-lingual learning. In doing so, we aim to answer the following research questions:

RQ 1 How can multitask learning be applied to the task of cross-lingual CWI?

RQ 2 How can complex words be identified in languages which are not seen during training time?

Our contributions also include a thorough qualitative and quantitative error analysis, which shows that long and infrequent words are very likely to be complex, but that non-complex words that display these properties pose a challenge to our system.

8.2 RELATED WORK

8.2.1 *Multitask Learning*

Multitask learning (MTL) is the combined learning of several tasks in a single model (Caruana, 1997). This can be beneficial in a number of scenarios. Previous work has shown benefits, e.g., in cases where one has tasks which are closely related to one another (Bjerva, 2017a,b), when one task can help another escape a local minimum (Bingel and Søgaard, 2017), and when one has access to some unsupervised signal which can be beneficial to the task at hand (Rei, 2017). A common approach to MTL is the application of hard parameter sharing, in which some set of parameters in a model is shared between several tasks. We contribute to previous work in MTL by using a hard parameter sharing approach in which we share intermediate layers between languages, and use one output-layer per language, thus in a sense seeing languages as tasks, similarly to Bjerva (2017a).

8.2.2 *Cross-lingual learning*

Cross-lingual learning is the problem of training a model on a given language, and applying it to another (unseen) language. One common approach is to apply cross-lingual word representations, though this has the disadvantage that it tends to place relatively high demands on availability of parallel text. Another frequently used approach in this context is to use machine translation (MT) so as to obtain a monolingual training set (Tiedemann et al., 2014). However, this approach necessarily increases the complexity of a system, as a fully-fledged MT system needs to be incorporated in the pipeline. Furthermore, this approach bypasses the problem of attempting to find methods or feature sets which can be successful across languages. We therefore follow previous work by, e.g. Bjerva and Östling (2017) in that we use hard parameter sharing with language-agnostic input representations. We build upon this by leveraging language-specific resources which are widely available, such as Wikipedia dumps, and WordNet (see Section 8.5.

8.2.3 *CWI*

Automatic complex word identification has a relatively short history as a research task, with first publications including Shardlow (2013a,b)

A noticeable commonality of the highest-scoring systems in the CWI 2016 shared task was the use of ensemble methods, most notably random forest classifiers, which drew on a range of morphologic, semantic and psycholinguistic features, among others (Paetzold and Specia, 2016c; Ronzano et al., 2016).

Yimam et al. (2017) present first work on CWI that considers languages other than English. They release a German and a Spanish dataset and present first CWI results for these languages. Notably, they also describe first cross-lingual experiments, in which they train on some language and test on another, i.e. without employing any of the common strategies for cross-lingual learning that we outline above.

Recently, Bingel et al. (2018b) showed promising results in predicting complex words from gaze patterns of Danish children with reading difficulties, which opens up possibilities for personalized complex word identification, but it is less certain how well their method generalizes to other languages or demographics.

8.3 DATA

We use the data made available through the shared task (Yimam et al., 2018). Each training instance consists of a sentence, with a marked complex phrase annotation, including the numbers of native and non-native annotators, and the fraction of these who found the phrase to be complex. An overview of the data is given in Table 8.1. The number of entries which are considered complex is quite skewed, and differs per language as French has substantially fewer complex phrases than English. This is further illustrated in Figure 8.1.

In addition to the shared task data, we also use external resources in our feature representations (see Section 8.5).

8.4 MODEL

As outlined in Section 8.2, earlier work has shown the aptitude of ensemble methods for CWI, especially such ensembles that feature random forests. We further choose to address the problem in a cross-lingual fashion, for which we deem multitask learning models particularly suitable.

Language	Training	Dev	Test	Complex
English	27,299	3,328	4,252	42.03%
Spanish	13,750	1,622	2,233	40.61%
German	6,151	795	959	39.21%
French	–	–	2,251	29.18%

Table 8.1: Data overview. The share of complex words is computed across all data splits.

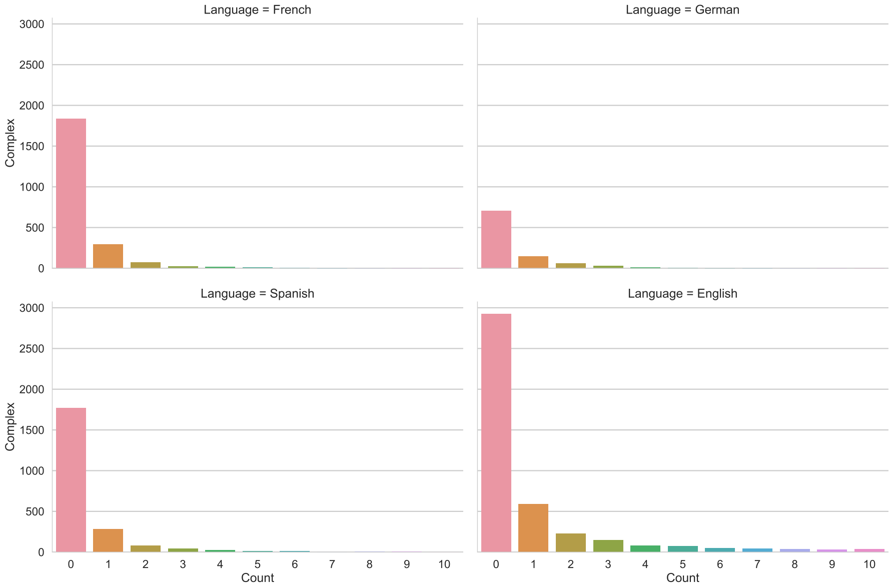


Figure 8.1: Histogram of numbers sentences (y-axis) which N annotators (x-axis) found to be complex.

Motivated by these observations, we devise an ensemble that comprises a number of random forests as well as feed-forward neural networks with hard parameter sharing. The random forests each consist of 100 trees that create splits based on Gini impurity (Breiman, 2001). They do not implement any form of explicit cross-lingual transfer other than the reliance on language-agnostic features, such that we simply train them on a single language at a time, or on shuffled concatenations of training data for several languages. We use random forest classifiers for the binary task and random-forest regressors for the probabilistic task. Note that our random forests are single-task models, where we cannot define shared or language-specific subparts. Instead, these are always trained on data for the single test language.

The neural MTL models, in contrast, explicitly define parts pertaining to specific languages. Concretely, for each language l, we define a function from a language-specific input layer to a hidden representation h_0 that we share between languages:

$$h_0 = \tanh(x^{(l)} W_{in}^{(l)} + b_{in}^{(l)}) \tag{8.1}$$

Here and in the following equations, $W_{(\cdot)}$ and $b_{(\cdot)}$ consistently denote weight matrices and bias vectors, respectively. $W_{in}^{(l)}$ and $b_{in}^{(l)}$ are the weights and bias terms specific to input layer l, and the input $x^{(l)}$ is a vector representation of the features introduced in Sec. 8.5.

We then compute deeper hidden representations, such that the hidden layer at depth d is defined as follows:

$$h_d = \tanh(h_{d-1} W_d + b_d) \tag{8.2}$$

Finally, each language l defines its own output $y^{(l)}$. This output is defined slightly differently for the regression and classification models.

$$y_{reg}^{(l)} = h_D W_{out}^{(l)} + b_{out}^{(l)} \tag{8.3}$$

For the former, this is simply a linear transformation of the deepest hidden layer D. The classification model adds a sigmoid activation to this:

$$y_{clf}^{(l)} = \sigma(h_D W_{out}^{(l)} + b_{out}^{(l)}) \tag{8.4}$$

8.4.1 *MTL training*

Since our multitask model defines several outputs, but our data is only labeled with a single annotation layer (i.e. for a single language or "task"), we need a training strategy that does not require true labels for all tasks. The way this is normally approached is to iteratively optimize for tasks in isolation, e.g. by deciding at random which language we sample a batch of data from at every pass of the forward-backward algorithm we use to train the model.

We employ the above strategy and optimize the regression model with a mean absolute error loss function, as well as cross-entropy for the classification model. We monitor these losses on the validation set as an early stopping criterion.

8.4.2 *Ensemble voting*

The different neural and random-forest based model that we train as devised above finally make independent predictions for new examples. For the regression models, we use the median prediction across all systems for a given input to make the final ensemble prediction. For the classifiers, however, we have an additional parameter t that we optimize on a held-out development set. This is a threshold indicating the fraction of classifiers that need to cast a positive vote for us to finally accept an example as complex. All neural and random forest classifiers are weighted equally here, each casting a single binary vote.

Lang.	MAE	Rank	Δ (system)	F_1	Rank	Δ (system)
French	0.066	1	0.012 (TMU)	0.7595	1	0.013 (TMU)
German	0.075	2	-0.013 (TMU)	0.6621	5	-0.083 (TMU)
Spanish	0.079	3	-0.007 (TMU)	0.7458	5	-0.024 (TMU)

Table 8.2: Official performance figures of our method for all non-English tracks. The Δ (system) column indicates the difference in performance between our system and the best system in each track except for ours. In accordance with the shared task report, classification performance is measured by macro F_1 between the complex and non-complex class in the official results.

8.4.3 *Language identification for cross-lingual prediction*

As we expect our system to be able to generate predictions for unseen languages (for which we have no explicit output layer), we implement a further component in our neural model that we optimize to predict the language of some input from the set of available languages with explicit output layers. This is an additional output layer of our model, defined as a dense projection from the first hidden layer followed by a sigmoid.

$$l = \sigma(h_0 W_{lid} + b_{lid}) \tag{8.5}$$

During training, we then supply a ground truth language identifier $\hat{l}$ as a second target variable and perform optimization under a cross-entropy loss that we add to the CWI loss. At test time, for a language without an explicit output layer, we first predict the most similar language we saw during training using Eq. 8.5 and then use the output layer for that language to generate CWI predictions. An alternative to this could be to generate predictions from all CWI output layers and ensemble these, possibly weighted, with weights inferred in a similar fashion to language identification.

For the random forest models, which do not define language-specific output functions, we simply concatenate training data from all available languages, leveraging the fact that our feature space is language-independent.

8.5 FEATURES

Our systems build on the same set of features for all input languages, although some of these are computed from language-specific resources. This means that the distributions of values attained for certain features may differ between languages, which is the motivation for language-specific input layers in our model. We further reduce language idiosyncrasies by normalizing all features to the $[0, 1]$ range. The features are listed below.

log-probability We compute unigram frequencies for candidate words as their log-probabilities in language-specific Wikipedia dumps. For multi-word targets, we use the sum of the log-probabilities of the individual items. Log-probabilities are computed using KenLM (Heafield, 2011).

character perplexity Based on the same Wikipedia dumps as above, we compute character perplexities over the candidate strings using a smoothed 5-gram character-based language model (again using KenLM).

number of synsets As a measure of a target's semantic ambiguity, we count the number of synsets that include it. For this, we rely on the language-specific WordNet resources for English (Fellbaum, 1998), Spanish (Gonzalez-Agirre et al., 2012) and French (Sagot and Fišer, 2008). For German, access to GermaNet (Hamp and Feldweg, 1997) was harder to obtain, and we instead automatically translate words from German to English and use the English WordNet.[38] In case of a multi-word target, we take the mean number of synsets across the individual words.

hypernym chain As a measure of semantic specificity, we further consider the length of the hypernym chain of an item, i.e. the number of hypernyms that can recursively be obtained for a word. These are also obtained using WordNet, and again we average over individual words in a target.

inflectional complexity As a proxy for inflectional complexity (i.e. the number of suffixes appended to a word stem), we measure the difference in length (character count) between the surface form and the stem of a word. For this, we use language-specific instances of the Snowball stemmer (Porter, 2001) as implemented in NLTK (Bird and Loper, 2004).

surface features As basic surface features, we include the length of an item (in characters) and whether it is all-lowercase.

bag-of-pos For each tag defined in the Universal Part-of-Speech project (Petrov et al., 2011), we count the number of words in a candidate that belong to the respective class. We obtain POS tags from spacy.[39]

target-sentence similarity Motivated by the conjecture that words or phrases are easier to understand if they display higher se-

[38] For the translations, we used a bilingual dictionary (`https://www.dict.cc/`).

[39] `https://spacy.io/`

mantic similarity with their context, we compute the cosine distances between averaged word embeddings for the target word or phrase and the rest of the containing sentence. To mitigate out-of-vocabulary problems, we use pretrained subword embeddings that we retrieve from the BPEemb project (Heinzerling and Strube, 2017).

The data provided for the shared task further includes information on how many of the annotators are native and non-native speakers. While this information is potentially helpful (assuming that non-native speakers would have a stronger bias for annotating as complex), we do not make use of it, considering that access to such information cannot be assumed in a real-world scenario.

8.6 RESULTS

We present an overview of the results that our method (as well as our best contender) achieved in Table 8.2 and discuss results for the individual languages below.[40]

8.6.1 *French*

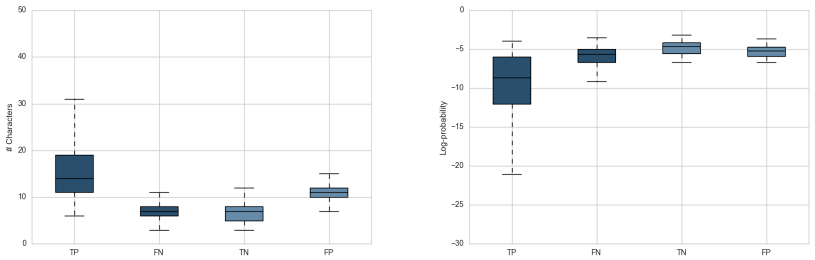
Due to the lack of training data for this track, it poses a challenging test for the ability of our models to generalize across languages. While the exact performance figures are at least partly subject to idiosyncrasies in the text samples and annotators, the results obtained here are a good benchmark of of what we can achieve for languages for which we do not even have validation data to monitor development loss for early stopping.

As Table 8.2 shows, we achieve the best results of all participating teams for French, both for the classification and for the regression track. We view this as evidence that our cross-lingual MTL approach is an effective means to share knowledge between different data sources and even different languages.

8.6.2 *German/Spanish*

Our results for Spanish and German show that, while we did not achieve the best results compared to other participants, our method still performs competitively. Especially for the regression track, while not ranking first, the absolute performance figures place us very close to the winning systems. We see this as a validation of our approach, in particular under the consideration that a gradual assessment of complexity is perhaps more meaningful than a binary one, especially

[40] We did not submit solutions for the English track.



(a) Length in characters per error type

(b) Log-probability per error type

Figure 8.2: Statistics of character length and language model log-probability for the French test set. The darker-shaded boxes are complex words that we predicted correctly (TP) and incorrectly (FN), respectively. The lighter-shaded boxes are non-complex words, predicted correctly (TN) and incorrectly (FP).

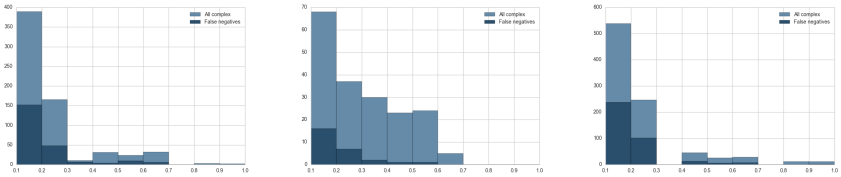


Figure 8.3: Distributions of false negative predictions per complexity degree as measured by the fraction of annotators that labeled items as complex in the French, German and Spanish test sets (left to right).

when the definition of the latter makes no distinction between one or all out of 20 annotators judging an item as difficult.

8.6.3 *Analysis*

QUALITATIVE ERROR ANALYSIS Table 8.3 exemplifies some of the correct and incorrect predictions that our system makes for the French test data. We observe that the system picks up on the relatively long targets listed as true positives. Note also that the false positives are relatively long words, which suggests that the system is deceived by this. The targets listed as false negatives are shorter, but they are examples of a (potentially unknown) named entity and a relatively technical term, which might pose difficulties to some readers. The words listed as true negatives are correctly predicted by our system as non-complex, possibly because of their shortness.

True positives
Il **marque néanmoins sa désapprobation** en voyant des Juifs prier devant le mur des Lamentations; Einstein commente qu'il s'agit de personnes collées au passé et faisant abstraction du présent.
Rimbaud a donné ses lettres de noblesse à un type de poème en prose distinct d'expériences plus **prosaïques** du type du “Spleen de Paris" de Baudelaire.
False negatives
Le pays des vallées d'Andorre entre la France et l'Espagne, sur le versant sud des **Pyrénées**, est constitué par deux vallées principales: celle du Valira del Orient et celle du Valira del Nord dont les eaux réunies forment le Valira.
Autres cultures permanentes, la lavande et le lavandin occupent plusieurs milliers d'**hectares** et fournissent plusieurs milliers d'emplois directs.
True negatives
Beaucoup d'îles des Caraïbes (les Antilles) – par exemple, les Grandes Antilles et les Petites Antilles – sont **situées** au-dessus de la plaque caraïbe, une plaque tectonique avec une topographie diffuse.
Avec un fort penchant à l'hermétisme qu'il partage avec d'autres de ses quasi contemporains (Gérard de Nerval, Stéphane Mallarmé, sinon Paul Verlaine parfois), Rimbaud a le **génie** des visions saisissantes qui semblent défier tout ordre de description du réel.
False positives
La **construction** de l'Atomium fut une prouesse technique.
La **proportion** des musulmans, tous sunnites, est inférieure à 1%.

Table 8.3: Example wins and losses of our model for French. Target words or phrases are marked in bold.

QUANTITATIVE ERROR ANALYSIS Investigating the observations from the previous section in a more quantitative fashion, Figure 8.2 presents distributions of two basic features across complex vs. non-complex words, and correctly vs. wrongly predicted test set items for French. For item length, we observe a clear pattern that complex words tend to be significantly longer than non-complex ones. Further, the longer they are, the easier it is for our model to detect them as complex. Non-complex words that are relatively long, however, lead to incorrect predictions from our model.

A very similar pattern can be observed for the log-probability of complex and non-complex items. The former are assigned a much lower probability by our language model, and particularly unlikely words are very easy to detect as complex. In turn, non-complex words with relatively low probability pose a challenge for our model.

FALSE NEGATIVES PER COMPLEXITY DEGREE We further analyze the influence of the degree of complexity on our model's ability to detect complex words. As stated in the Introduction, an item is labeled as complex in the classification setting if any of the annotators deemed it to be complex. Effectively, no distinction is made in the classification task between a "slightly complex" item that was marked as such by just one out of ten annotators, and one that was unanimously considered complex.

A natural assumption is that our models would more easily pick up on highly complex words and predict false negatives mostly for items with low complex annotation ratios. To verify this assumption, we plot the total number of complex words in the three non-English test sets against the false negative predictions of our model, grouped by the ratio of annotators who marked an item as complex (Figure 8.3). The French and Spanish test sets are somewhat inconclusive for our question as they generally contain very few items with a complexity ratio higher than 0.2. The German test set, however, is more balanced, and in fact we observe that items with a complexity ratio above 0.2 are very reliably detected by our model, confirming our hypothesis.

8.7 DISCUSSION

We approached **RQ 1** by using one output layer per language, and sharing intermediate parameters. This approach was successful, at least in part due to our language-agnostic input representations, which allowed the model to learn similar internal representations for each language. Separate output-layers per language, in turn, allow for the model to make language-specific accommodations. We approached **RQ 2** by using language-agnostic feature representations, and language-specific output layers which were chosen during test time for unseen languages. This approach allowed our model to perform well on the unseen language French, and in fact outperformed our results on other languages. This is, however, not strictly a fair comparison as it is possible that the French test set was somehow easier than the others.

8.8 CONCLUSION

We tackled the 2018 Shared Task on CWI with a cross-lingual approach via multitask learning. Our system is highly language-agnostic, as evidenced by our high performance on French, which was not seen during training time. Our analysis confirms that word length and frequency are good, cross-linguistic predictors of complexity. However, the concrete relationships between these features and complexity may differ between languages, which is captured by our multitask learning approach. Our approach is especially promising for the application of CWI to unseen languages, as we do not assume access to any target language training data. Furthermore, this could even substantially facilitate the creation of new CWI datasets, using a bootstrapping or active learning approach.

ACKNOWLEDGMENTS

James Thorne wrote much of the Pytorch code that we used in this work. We are grateful to the anonymous reviewer for helpful suggestions. The first author acknowledges the support by Trygfonden.

Part V

CONCLUSIONS

9 DISCUSSION OF THE CONTRIBUTIONS

The previous chapters of this thesis have introduced work on adaptive and personalized text simplification. The papers mark individual contributions to this overarching topic and address various aspects of the central research question posed on page 10:

> *How can we make simplification more adaptive and personalizable for the individual end user?*

The considerations that are discussed in the introductory chapter devise a modification of the simplification workflow that allows for the integration of user preferences and feedback, such that a simplification model can base its decisions on what it knows about a user's individual needs and desires. To conclude this thesis, let us begin by reviewing how these papers contribute towards this goal and how they consequently help answer the research question stated above.

ADAPTABLE SENTENCE SIMPLIFICATION

The papers in chapters 3 and 4 present approaches to sentence-level simplification that can accommodate user preferences and have the capacity to adapt their final output accordingly. In explicitly predicting available simplification operations locally at the word or phrase level, user preferences can be incorporated by promoting or suppressing certain simplification operations altogether or in specific contexts. A user may, for example, opt to have the system shorten sentences longer than 15 words through phrase deletions, but not perform lexical substitution.

In the first of these two chapters, we present an approach that learns deletions and paraphrases over dependency trees (Bingel and Søgaard, 2016). The application of simplification operations to entire subtrees is mainly motivated by grammaticality considerations, assuming that dependency subtrees correspond to syntactic units and that failing to remove or change them in their entirety will likely cause ungrammatical output with incomplete phrases. Another benefit of our method is that it relies on training data that is several orders of magnitude smaller than that of similar, state-of-the-art compression approaches. In a human evaluation, we find that our approach produces better readable output sentences than previous approaches that tackle compression and paraphrasing jointly, while being comparable to state-of-the-art approaches to compression only.

The second work that addresses adaptable sentence simplification does so by learning and predicting explicit simplification operations in a sentence (Alva-Manchego et al., 2017). Since such explicit operations are not manually annotated in existing simplification corpora, our approach heuristically infers them from word alignments between pairs of original and simplified sentences. Human evaluation demonstrates that, applying all available simplification operations, both of the above approaches produce simpler output than strong baselines, as judged by non-native speakers of English.

PERSONALIZED LEXICAL SIMPLIFICATION

As the second major line of work, this thesis has presented text simplification methods that learn directly from user feedback, rather than requiring the user to explicitly turn knobs and dials to let the model know their personal simplification needs and preferences. This is motivated, in part, by the conjecture that users do not necessarily have a sufficient level of introspection into their own specific needs. Further, it is impractical to devise and implement sufficiently detailed set screws that let the user configure and express minuscule details and features that contribute to, for instance, perceived word difficulty. Work on self-adapting and personalizing lexical simplification is covered by chapters 5 and 6.

In chapter 5 (Bingel et al., 2018b), we investigate ways to determine word difficulty based on individual gaze patterns in children with reading difficulties. In particular, our aim is to predict words in context that are decoded incorrectly while reading out loud. We demonstrate that eye-tracking data contains a strong signal for the individual perception of word difficulty, letting us predict misread words with high precision. Another finding is the heterogeneity of readers, attested by the largely different results between users and the difficulty in sharing information between them. Notably, the eye-tracking data we use in this study comes from noisy, real-world gaze recordings of children's reading. This is in contrast to most scientific experiments involving eye-tracking, where the data is typically recorded in highly idealized lab settings, including identical or controlled stimuli, lighting conditions, and subjects, among others. The contributions of this study thus include the demonstration of the fitness of our method despite these complicated conditions for data collection.

Another contribution to self-adapting lexical simplification is presented in chapter 6 (Bingel et al., 2018a). Here, we motivate, devise and implement a software that performs individual lexical simplifications in a browser, keeping separate user accounts which continuously personalize as the user receives and rates simplifications from the system. The system implements the established lexical simplification pipeline (Shardlow, 2014b), but adapts it to accommodate user feedback in an online learning fashion. Concretely, the substitution ranking step in the pipeline is implemented as a ranking problem between two candidates at a time. A model is thus initialized from a static corpus of binary rankings, then user feedback is integrated by determining the preferred of two options and feeding this information back to the online learning function. With time, the model will learn a user's individual simplification needs and preferences. Besides serving as a practical application that promotes accessibility to text, the software will enable us to perform further analyses on a range of research questions, including the investigation of individual reading profiles, and invariants between them or between groups of readers, among others.

Evidence for the effectiveness of adaptive solutions to lexical simplification has very recently come from the works of Yimam and Biemann (2018) and Lee and Yeung (2018). These papers demonstrate that a small number of training examples annotated by a specific individual can lead to significantly better personalized simplification models, compared to generic solutions. In particular, Yimam and Biemann investigate a scenario in which a user provides feedback over several rounds of model adaptation, finding that even after a few rounds the model understands the user's individual simplification needs much better than the base system.

MULTI-TASK LEARNING FOR SIMPLIFICATION

The two last papers included in this thesis present work on multi-task learning, a framework that in recent years has gained increasing attention in many application areas of natural language processing. Multitask learning essentially lets a system learn several functions at once, sharing information between these through joint parameters or loss functions, for instance. This may lead to better generalization through regularization or the induction of certain biases. Further, MTL allows us to integrate information from several, disparate datasets at once, which can overcome some of the challenges in simplification revolving around data scarcity.

However, the questions *why* and *when* MTL works had remained largely unanswered for some time. Chapter 7 thus investigates the conditions under which multi-task learning can be expected to lead to benefits (Bingel and Søgaard, 2017). Comparing performance changes between single-task and multi-task models across ten different sequence labeling tasks, we find that MTL leads to improvements in just under half the cases. We then test a range of 42 task descriptors for their ability to predict MTL gains, using them as features in a logistic regression model. Here, we find that the number of labels as well as their entropy in the used dataset are good predictors of gains from multi-task setups. Specifically, if the main task has many different labels (and is therefore difficult to learn), MTL gains become more likely. Even more clearly, we see that MTL typically leads to gains when the main task plateaus quickly in a single-task learning setup, suggesting that MTL can help difficult tasks escape local optima.

In the context of simplification, efforts to share information between individual readers as approached in chapter 5 have so far been largely unsuccessful. In chapter 8 (Bingel and Bjerva, 2018), however, we tackle complex word identification in a cross-lingual setting, sharing information between different *languages* to increase performance. This approach also generalizes to languages for which no data was seen at training time. In simplification research, this is an encouraging finding, considering that annotated training data does not exist for most languages. This contribution marks the first successful application of multi-task learning to a simplification task.

10 OUTLOOK

Finally, let us shed light on some of the most prominent points of departure for future research that this thesis offers. The previous chapter has summarized the contributions of this thesis and has grouped them into three major lines of work. A natural next direction of research will be to connect the dots between these. In particular, the development and investigation of simplification models that are at the same time customizable through explicit input as well as self-personalizing from implicit feedback will be a logical continuation of the research presented here, and would get us even closer to the augmented simplification workflow devised in the introductory chapter (see Figure 1.2).

Another line of future work relates to further evaluating the presented approaches out in the open. From the argumentation in the first chapter, it follows that simplification systems ultimately need to be evaluated in real-world settings on an individual basis. With respect to applications that go beyond in-vitro lab settings, we have already covered some ground, for example with the work presented in chapter 5, which is based on data collected in naturalistic reading scenarios and can be directly integrated into the software that was used to collect the data. Additionally, chapter 6 describes a lexical simplification system that was officially released shortly before the completion and publication of this thesis and is freely available to the public. The software, Lexi, enables large-scale user studies to evaluate the effectiveness of the proposed online-learning method. Such studies are a further candidate for work following this thesis.

Along similar lines, Lexi allows for user studies that can give further insight into a potential typology of readers and reading strategies. Originally, the basic assumption of heterogeneous readers with individual reading difficulties is licensed through existing theory and a limited body of empirical evidence. Chapter 5 adds to this body, but a natural development of our line of work will give further support to this hypothesis.

Lastly, text simplification models generally face a number of challenges, the most prominent of which we discuss in chapter 2. Much work is needed in this direction, in particular to safeguard the grammaticality and meaning preservation of simplification output.

Part VI

APPENDIX

A

SUPPLEMENTARY MATERIAL FOR INDIVIDUAL STUDIES

A.1 EXAMPLES OF JSON PROTOCOL USED BY LEXI

A.1.1 *Request sent from frontend to backend*

```
{
  'frontend_version': '1.0.0',
  'user': 'lexi-user@mail.com',
  'html': '<p>Natasja startede allerede som 13-årig med at
      synge og DJ\'e  ... </p>',
  'startOffset': 17,
  'endOffset': 25,
  'url': 'https://da.wikipedia.org/wiki/Natasja'
}
```

A.1.2 *Reply from backend to frontend*

```
{
  'backend_version': '1.0.0',
  'html': '<p>Natasja startede <span id="lexi_254_1" class="
      lexi-simplify">allerede</span> som 13-årig med at
      synge og DJ\'e ... </p>',
  'session_id': 254,
  'status': 200,
  'message': 'Simplification successful',
  'simplifications': {
    'lexi_254_1': {
       'choices': ['allerede', 'bare'],
       'selection': 0,
       'sentence': "Natasja startede allerede som 13-årig
           med at synge og DJ'e i København, hvor hun gjorde
            sig bemærket sammen med Miss Mukupa, McEmzee og
           DJ Kruzh'em i bandet No Name Requested.",
       'word_index': 2
    }
  }
}
```

A.1.3 *Feedback sent from frontend to backend*

```
{
  'frontend_version': '1.0.0',
  'user': 'lexi-user@mail.com',
  'html': '<p>Natasja startede <span id="lexi_254_1" class="
      lexi-simplify">bare</span> som 13-årig med at synge og
       DJ\'e ...
     </p>',
  'session_id': 254,
  'simplifications': {
    'lexi_254_1': {
      'choices': ['allerede', 'bare'],
      'selection': 1,
      'sentence': "Natasja startede allerede som 13-årig med
          at synge
        og DJ'e i København, hvor hun gjorde sig bemærket
           sammen med
        Miss Mukupa, McEmzee og DJ Kruzh'em i bandet No Name
        Requested.",
      'word_index': 2
    },
  },
  'rating': 4,
  'url': 'https://da.wikipedia.org/wiki/Natasja',
  'session_id': 254
}
```

A.2 ALGORITHMS USED FOR TRAINING DATA GENERATION IN STUDY "HOW TO SIMPLIFY FROM EXPLICIT LABELING OF COMPLEX-SIMPLIFIED TEXT PAIRS"

Algorithmus 1 : Initial word-level annotation

Input : O: list with tokens of the original sentence, S: list with tokens of the simplified sentence, A: list with word alignments.

Output : SLO: simplification labels for each token in O, SLS: simplification labels for each token in S.

```
// labeling tokens in the original sentence
for $i \leftarrow 1$ to $\mathrm{len}(O)$ do
    // get the indexes of the tokens in S to which the ith
       token in O is aligned to
    $IS \leftarrow$ FindAlignments($A, i, 's'$)
    if $\mathrm{len}(IS) > 0$ then                              // it is aligned
        if $\mathrm{len}(IS) = 1$ and $O_i = S_{IS_0}$ then
            $SLO_i \leftarrow$ 'C'                               // keep
        else                                          // not an exact match
            $SLO_i \leftarrow$ 'R'                               // replace
        end
    else                                                    // not aligned
        $SLO_i \leftarrow$ 'D'                                   // delete
    end
end
// labeling tokens in the simplified sentence
for $j \leftarrow 1$ to $\mathrm{len}(S) + 1$ do
    // get the indexes of the tokens in O to which the jth
       token in S is aligned to
    $IO \leftarrow$ FindAlignments($A, j, 'o'$)
    if $\mathrm{len}(IO) > 0$ then                              // it is aligned
        $SLS_j \leftarrow$ 'O'
        if $\mathrm{len}(IO) > 1$ then
            // the current token in S replaces a phrase in O
            foreach $k \in IO$ do
                $SLO_k \leftarrow$ 'R'
            end
        end
    else
        $SLS_j \leftarrow$ 'A'                                   // add
    end
end
```

Algorithmus 2 : Annotation of reorderings

```
Input : SLO: simplification labels for each token in original sentence,
        SLS: simplification labels for each token in simplified
        sentence, $A$: list with word alignments.
Output : SLO modified.
shift_left $\leftarrow 0$
for $i \leftarrow 0$ to $\mathrm{len}(SLO)$ do
    if $SLO_i \in$ ['D', 'R'] then
        shift_left $\leftarrow$ shift_left $+ 1$
    else
        $IS \leftarrow$ FindAlignments($A$, $i$, 's')
        if $\mathrm{len}(IS) > 0$ then
            $k \leftarrow IS_0$        // index of the aligned token in the
             simplified sentence
        else
            $k \leftarrow i$        // index of the token in the original
             sentence
        end
        shift_right $\leftarrow 0$
        for $j \leftarrow 0$ to $k$ do
            if $SLS_j \in$ ['AC', 'RW'] then
                shift_right $\leftarrow$ shift_right $+ 1$
            end
        end
        if $i -$ shift_left $+$ shift_right $\neq k$ then
            switch $SLS_i$ do
                case 'C' do $SLS_i \leftarrow$ 'M'
                case 'R' do $SLS_i \leftarrow$ 'RM'
                case 'RW' do $SLS_i \leftarrow$ 'RWM'
            end
        end
    end
end
```

BIBLIOGRAPHY

Alexander, Regi, Peter E Langdon, Verity Chester, Magali Barnoux, Ignatius Gunaratna, and Sudeep Hoare (2016). “Heterogeneity within autism spectrum disorder in forensic mental health: the introduction of typologies.” In: *Advances in Autism* 2.4, pp. 201–209.

Alva-Manchego, Fernando, Joachim Bingel, Gustavo H. Paetzold, Carolina Scarton, and Lucia Specia (2017). “Learning How to Simplify From Explicit Labeling of Complex-Simplified Text Pairs.” In: *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Vol. 1, pp. 295–305.

Anderson, Richard (1981). “A proposal to continue a center for the study of reading.” In: *Urbana: University of Illinois*.

Azab, Mahmoud, Chris Hokamp, and Rada Mihalcea (2015). “Using Word Semantics To Assist English as a Second Language Learners.” In: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*. Denver, Colorado: Association for Computational Linguistics, pp. 116–120. URL: http://www.aclweb.org/anthology/N15-3024.

Baeza-Yates, Ricardo, Luz Rello, and Julia Dembowski (2015). “CASSA: A Context-Aware Synonym Simplification Algorithm.” In: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pp. 1380–1385.

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio (2014). “Neural machine translation by jointly learning to align and translate.” In: *arXiv preprint arXiv:1409.0473*.

Bakker, Dirk J (1992). “Neuropsychological classification and treatment of dyslexia.” In: *Journal of learning disabilities* 25.2, pp. 102–109.

Barrett, Maria, Joachim Bingel, Frank Keller, and Anders Søgaard (2016). “Weakly supervised part-of-speech tagging using eye-tracking data.” In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 579–584.

Barzilay, Regina and Noemie Elhadad (2003). “Sentence alignment for monolingual comparable corpora.” In: *Proceedings of EMNLP*, pp. 25–32. URL: https://doi.org/10.3115/1119355.1119359.

Baxter, Jonathan (2000). “A model of inductive bias learning.” In: *Journal of Artificial Intelligence Research* 12, pp. 149–198.

Beatty, Jackson, Brennis Lucero-Wagoner, et al. (2000). “The pupillary system.” In: *Handbook of psychophysiology* 2, pp. 142–162.

Ben-David, Shai and Reba Schuller Borbely (2008). "A notion of task relatedness yielding provable multiple-task learning guarantees." In: *Machine learning* 73.3, pp. 273–287.

Bender, Emily M., Leon Derczynski, and Pierre Isabelle, eds. (2018). *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*. Association for Computational Linguistics. ISBN: 978-1-948087-50-6. URL: https://aclanthology.info/volumes/proceedings-of-the-27th-international-conference-on-computational-linguistics.

Benfatto, Mattias Nilsson, Gustaf Öqvist Seimyr, Jan Ygge, Tony Pansell, Agneta Rydberg, and Christer Jacobson (2016). "Screening for dyslexia using eye tracking during reading." In: *PloS one* 11.12, e0165508.

Benton, Adrian, Margaret Mitchell, and Dirk Hovy (2017). "Multitask Learning for Mental Health Conditions with Limited Social Media Data." In: *EACL*.

Bernth, Arendse (1998). "EasyEnglish: Preprocessing for MT." In: *Proceedings of the Second International Workshop on Controlled Language Applications*, pp. 30–41.

Bingel, Joachim and Johannes Bjerva (2018). "Cross-lingual complex word identification with multitask learning." In: *Proceedings of the Complex Word Identification Shared Task at the 13th Workshop on Innovative Use of NLP for Building Educational Applications*. New Orleans, United States: Association for Computational Linguistics.

Bingel, Joachim and Anders Søgaard (2016). "Text simplification as tree labeling." In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Vol. 2, pp. 337–343.

Bingel, Joachim and Anders Søgaard (2017). "Identifying beneficial task relations for multi-task learning in deep neural networks." In: *15th Conference of the European Chapter of the Association for Computational Linguistics*.

Bingel, Joachim, Gustavo Paetzold, and Anders Søgaard (2018a). "Lexi: a tool for adaptive, personalized text simplification." In: *COLING*, pp. 164–169.

Bingel, Joachim, Maria Barrett, and Sigrid Klerke (2018b). "Predicting misreadings from gaze in children with reading difficulties." In: *Proceedings of the 13th Workshop on Innovative Use of NLP for Building Educational Applications*. New Orleans, United States: Association for Computational Linguistics.

Biran, Or, Samuel Brody, and Noémie Elhadad (2011). "Putting it simply: a context-aware approach to lexical simplification." In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*. Association for Computational Linguistics, pp. 496–501.

Bird, Steven and Edward Loper (2004). "NLTK: the natural language toolkit." In: *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*. Association for Computational Linguistics, p. 31.

Bjerva, Johannes (2017a). "One Model to Rule them all: Multitask and Multilingual Modelling for Lexical Analysis." PhD thesis. University of Groningen.

Bjerva, Johannes (2017b). "Will my auxiliary tagging task help? Estimating Auxiliary Tasks Effectivity in Multi-Task Learning." In: *Proceedings of the 21st Nordic Conference on Computational Linguistics, NoDaLiDa, 22-24 May 2017, Gothenburg, Sweden*. 131. Linköping University Electronic Press, Linköpings universitet., pp. 216–220.

Bjerva, Johannes and Robert Östling (2017). "Cross-lingual Learning of Semantic Textual Similarity with Multilingual Word Representations." In: *Proceedings of the 21st Nordic Conference on Computational Linguistics, NoDaLiDa, 22-24 May 2017, Gothenburg, Sweden*. 131. Linköping University Electronic Press, Linköpings universitet, pp. 211–215.

Björnsson, Carl Hugo (1968). *Läsbarhet*. Liber.

Blythe, Hazel I and Holly SSL Joseph (2011). "Children's eye movements during reading." In:

Bojanowski, Piotr, Edouard Grave, Armand Joulin, and Tomas Mikolov (2016). "Enriching Word Vectors with Subword Information." In: *arXiv preprint arXiv:1607.04606*.

Bollman, Marcel and Anders Søgaard (2016). "Improving historical spelling normalization with bi-directional LSTMs and multi-task learning." In: *COLING*.

Bott, Stefan and Horacio Saggion (2011). "An unsupervised alignment algorithm for text simplification corpus construction." In: *Proceedings of MTTG*, pp. 20–26. URL: http://www.aclweb.org/anthology/W11-1603.

Bott, Stefan, Luz Rello, Biljana Drndarevic, and Horacio Saggion (2012a). "Can spanish be simpler? lexsis: Lexical simplification for spanish." In: *Proceedings of COLING 2012*, pp. 357–374.

Bott, Stefan, Horacio Saggion, and Simon Mille (2012b). "Text Simplification Tools for Spanish." In: *LREC*, pp. 1665–1671.

Braud, Chloe, Barbara Plank, and Anders Søgaard (2016). "Multiview and multi-task training of RST discourse parser." In: *COLING*.

Breiman, Leo (2001). "Random forests." In: *Machine learning* 45.1, pp. 5–32.

Brinton, Laurel J and Donna Brinton (2010). *The linguistic structure of modern English*. John Benjamins Publishing.

Brunswick, Nicola (2010). "Unimpaired reading development and dyslexia across different languages." In: *Reading and dyslexia in different orthographies*. Psychology Press, pp. 149–172.

Buringh, Eltjo and Jan Luiten Van Zanden (2009). "Charting the "Rise of the West": Manuscripts and Printed Books in Europe, a long-term Perspective from the Sixth through Eighteenth Centuries." In: *The Journal of Economic History* 69.2, pp. 409–445.

Burns, Philip R (2013). "Morphadorner v2: A java library for the morphological adornment of english language texts." In: *Northwestern University, Evanston, IL.*

Burstein, Jill, Jane Shore, John Sabatini, Yong-Won Lee, and Matthew Ventura (2007). "The automated text adaptation tool." In: *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*. Association for Computational Linguistics, pp. 3–4.

Canning, Yvonne and John Tait (1999). "Syntactic simplification of newspaper text for aphasic readers." In: *ACM SIGIR'99 Workshop on Customised Information Delivery*, pp. 6–11.

Canning, Yvonne, John Tait, Jackie Archibald, and Ros Crawley (2000). "Replacing anaphora for readers with acquired dyslexia." In: *Proceedings of the 4th Discourse Anaphora and Anaphor Resolution Colloquium (DAARC'00), Lancaster, UK.*

Carrillo, MS, J Alegrıa, P Miranda, and Sánchez Pérez (2011). "Evaluación de la dislexia en la escuela primaria: Prevalencia en espanol (Evaluation of dyslexia in primary school: The prevalence in Spanish)." In: *Escritos de Psicologıa (Psychology Writings)* 4.2, pp. 35–44.

Carroll, John, Guido Minnen, Yvonne Canning, Siobhan L. Devlin, and John Tait (1998). "Practical simplification of English newspaper text to assist aphasic readers." In: *Proceedings of the AAAI-98 Workshop on Integrating Artificial Intelligence and Assistive Technology*, pp. 7–10.

Carroll, John, Guido Minnen, Darren Pearce, Yvonne Canning, Siobhan Devlin, and John Tait (1999). "Simplifying text for language-impaired readers." In: *Proceedings of EACL*. Vol. 99, pp. 269–270.

Caruana, Rich (1993). "Multitask learning: a knowledge-based source of inductive bias." In: *ICML*.

Caruana, Rich (1997). "Multitask learning." In: *Machine Learning* 28 (1), pp. 41–75.

Chandrasekar, Raman and Bangalore Srinivas (1997). "Automatic induction of rules for text simplification1." In: *Knowledge-Based Systems* 10.3, pp. 183–190.

Chandrasekar, Raman, Christine Doran, and Bangalore Srinivas (1996). "Motivations and methods for text simplification." In: *Proceedings of the 16th conference on Computational linguistics-Volume 2*. Association for Computational Linguistics, pp. 1041–1044.

Cho, Kyunghyun, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio (2014). "On the properties of neural machine translation: Encoder-decoder approaches." In: *arXiv preprint arXiv:1409.1259*.

Chodorow, Martin and Claudia Leacock (2000). "An unsupervised method for detecting grammatical errors." In: *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*. Association for Computational Linguistics, pp. 140–147.

Chollet, François (2015). *Keras*. https://github.com/fchollet/keras.

Chopra, Sumit, Michael Auli, and Alexander M Rush (2016). "Abstractive sentence summarization with attentive recurrent neural networks." In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 93–98.

Clarke, James and Mirella Lapata (2008). "Global inference for sentence compression: An integer linear programming approach." In: *Journal of Artificial Intelligence Research*, pp. 399–429.

Cohn, Trevor and Mirella Lapata (2008). "Sentence compression beyond word deletion." In: *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*. Association for Computational Linguistics, pp. 137–144.

Cohn, Trevor and Mirella Lapata (2009). "Sentence compression as tree transduction." In: *Journal of Artificial Intelligence Research*, pp. 637–674.

Collins-Thompson, Kevyn (2014). "Computational assessment of text readability: A survey of current and future research." In: *ITL-International Journal of Applied Linguistics* 165.2, pp. 97–135.

Collobert, Ronan and Jason Weston (2008). "A unified architecture for natural language processing: Deep neural networks with multitask learning." In: *ICML*.

Coster, William and David Kauchak (2011a). "Learning to Simplify Sentences Using Wikipedia." In: *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, pp. 1–9. URL: http://www.aclweb.org/anthology/W11-1601.

Coster, William and David Kauchak (2011b). "Learning to simplify sentences using wikipedia." In: *Proceedings of the workshop on monolingual text-to-text generation*. Association for Computational Linguistics, pp. 1–9.

Coster, William and David Kauchak (2011c). "Simple English Wikipedia: a new text simplification task." In: *Proceedings of ACL*, pp. 665–669. URL: http://www.aclweb.org/anthology/P11-2117.

Cummins, Ronan and Marek Rei (2018). "Neural Multi-task Learning in Automated Assessment." In: *arXiv preprint arXiv:1801.06830*.

Dale, Edgar and Jeanne S Chall (1948). "A formula for predicting readability: Instructions." In: *Educational research bulletin*, pp. 37–54.

De Belder, Jan and Marie-Francine Moens (2010). "Text simplification for children." In: *Prroceedings of the SIGIR workshop on accessible search systems*. ACM, pp. 19–26.

De Belder, Jan, Koen Deschacht, and Marie-Francine Moens (2010). "Lexical simplification." In: *Proceedings of ITEC2010: 1st international conference on interdisciplinary research on technology, education and communication*.

Devlin, Siobhan L. (1999). "Simplifying natural language for aphasic readers." PhD thesis. University of Sunderland.

Devlin, Siobhan L. and John Tait (1998). "The use of a psycholinguistic database in the simplification of text for aphasic readers." In: *Linguistic databases*.

Devlin, Siobhan L. and Gary Unthank (2006). "Helping aphasic people process online information." In: *Proceedings of the 8th SIGACCESS*, pp. 225–226.

Dras, Mark (1999). *Tree adjoining grammar and the reluctant paraphrasing of text*. Macquarie University Sydney.

Drndarevic, Biljana and Horacio Saggion (2012). "Towards Automatic Lexical Simplification in Spanish: An Empirical Study." In: *Proceedings of the First Workshop on Predicting and Improving Text Readability for target reader populations*. Montrèal, Canada: Association for Computational Linguistics, pp. 8–16. URL: http://www.aclweb.org/anthology/W12-2202.

Duchi, John, Elad Hazan, and Yoram Singer (2011). "Adaptive subgradient methods for online learning and stochastic optimization." In: *Journal of Machine Learning Research* 12, pp. 2121–2159. URL: http://www.jmlr.org/papers/volume12/duchi11a/duchi11a.pdf.

Elming, Jakob, Anders Johannsen, Sigrid Klerke, Emanuele Lapponi, Hector Martinez Alonso, and Anders Søgaard (2013). "Downstream effects of tree-to-dependency conversions." In: *HLT-NAACL*, pp. 617–626.

Evans, Richard, Constantin Orasan, and Iustin Dornescu (2014). "An evaluation of syntactic simplification rules for people with autism." In: *Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations (PITR)*, pp. 131–140.

Feblowitz, Dan and David Kauchak (2013). "Sentence Simplification as Tree Transduction." In: *Proceedings of the PITR Workshop*, pp. 1–10. URL: http://www.aclweb.org/anthology/W13-2901.

Fellbaum, Christiane (1998). "A semantic network of English verbs." In: *WordNet: An electronic lexical database* 3, pp. 153–178.

Ferrés, Daniel, Horacio Saggion, and Xavier Gómez Guinovart (2017). "An Adaptable Lexical Simplification Architecture for Major Ibero-Romance Languages." In: *Proceedings of the First Workshop on Building Linguistically Generalizable NLP Systems*. Copenhagen, Denmark:

Association for Computational Linguistics, pp. 40–47. URL: http://aclweb.org/anthology/W17-5406.

Fielding, Roy Thomas (2000). "REST: architectural styles and the design of network-based software architectures." In: *Doctoral dissertation, University of California*.

Filippova, Katja and Yasemin Altun (2013). "Overcoming the Lack of Parallel Data in Sentence Compression." In: *EMNLP*, pp. 1481–1491.

Filippova, Katja and Michael Strube (2008). "Dependency tree based sentence compression." In: *Proceedings of the Fifth International Natural Language Generation Conference*. Association for Computational Linguistics, pp. 25–32.

Filippova, Katja, Enrique Alfonseca, Carlos A Colmenares, Lukasz Kaiser, and Oriol Vinyals (2015). "Sentence Compression by Deletion with LSTMs." In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 360–368.

Flesch, Rudolph (1948). "A new readability yardstick." In: *Journal of applied psychology* 32.3, p. 221.

Gamon, Michael (2011). "High-order sequence modeling for language learner error detection." In: *Proceedings of the 6th Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics, pp. 180–189.

Ganitkevitch, Juri, Benjamin Van Durme, and Chris Callison-Burch (2013). "PPDB: The Paraphrase Database." In: *Proceedings of NAACL-HLT*. Atlanta, Georgia: Association for Computational Linguistics, pp. 758–764. URL: http://cs.jhu.edu/~ccb/publications/ppdb.pdf.

Gasperin, Caroline, Lucia Specia, Tiago Pereira, and Sandra Aluísio (2009). "Learning when to simplify sentences for natural text simplification." In: *Proceedings of ENIA*, pp. 809–818.

Gibson, Edward and James Thomas (1999). "Memory limitations and structural forgetting: The perception of complex ungrammatical sentences as grammatical." In: *Language and Cognitive Processes* 14.3, pp. 225–248.

Glavaš, Goran and Sanja Štajner (2015). "Simplifying Lexical Simplification: Do We Need Simplified Corpora?" In: *Proceedings of the 53rd ACL*. Beijing, China: Association for Computational Linguistics, pp. 63–68. URL: http://www.aclweb.org/anthology/P15-2011.

Gonzalez-Agirre, Aitor, Egoitz Laparra, and German Rigau (2012). "Multilingual Central Repository version 3.0." In: *LREC*, pp. 2525–2529.

Gonzalez-Garduño, Ana Valeria and Anders Søgaard (2017). "Using gaze to predict text readability." In: *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pp. 438–443.

Grosz, Barbara J, Scott Weinstein, and Aravind K Joshi (1995). "Centering: A framework for modeling the local coherence of discourse." In: *Computational linguistics* 21.2, pp. 203–225.

Hading, Muhaimin, Yuji Matsumoto, and Maki Sakamoto (2016). "Japanese Lexical Simplification for Non-Native Speakers." In: *Proceedings of the 3rd Workshop on Natural Language Processing Techniques for Educational Applications (NLPTEA2016)*. Osaka, Japan: The COLING 2016 Organizing Committee, pp. 92–96. URL: http://www.aclweb.org/anthology/W16-4912.

Hamp, Birgit and Helmut Feldweg (1997). "Germanet-a lexical-semantic net for German." In: *Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*.

Heafield, Kenneth (2011). "KenLM: Faster and smaller language model queries." In: *Proceedings of the Sixth Workshop on Statistical Machine Translation*. Association for Computational Linguistics, pp. 187–197.

Heinzerling, Benjamin and Michael Strube (2017). "BPEmb: Tokenization-free Pre-trained Subword Embeddings in 275 Languages." In: *arXiv preprint arXiv:1710.02187*.

Hockenmaier, Julia and Mark Steedman (2007). "CCGbank: A Corpus of CCG Derivations and Dependency Structures Extracted from the Penn Treebank." In: *Comput. Linguist.* 33.3, pp. 355–396. ISSN: 0891-2017. DOI: 10.1162/coli.2007.33.3.355. URL: http://dx.doi.org/10.1162/coli.2007.33.3.355.

Horn, Colby, Cathryn Manduca, and David Kauchak (2014). "Learning a lexical simplifier using Wikipedia." In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Vol. 2, pp. 458–463.

Hwang, William, Hannaneh Hajishirzi, Mari Ostendorf, and Wei Wu (2015). "Aligning sentences from standard wikipedia to simple wikipedia." In: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 211–217.

Hyönä, Jukka and Richard K Olson (1995). "Eye fixation patterns among dyslexic and normal readers: Effects of word length and word frequency." In: *Journal of Experimental Psychology: Learning, Memory, and Cognition* 21.6, pp. 1430–40.

Interagency Committee on Learning Disabilities (1987). *Learning Disabilities: A Report to the U.S. Congress.* Tech. rep. Government Printing Office, Washington DC, U.S.

Inui, Kentaro, Atsushi Fujita, Tetsuro Takahashi, Ryu Iida, and Tomoya Iwakura (2003). "Text simplification for reading assistance: a project note." In: *Proceedings of the second international workshop on Paraphrasing-Volume 16*. Association for Computational Linguistics, pp. 9–16.

Joseph, Holly SSL, Simon P Liversedge, Hazel I Blythe, Sarah J White, Susan E Gathercole, and Keith Rayner (2008). "Children's and adults' processing of anomaly and implausibility during reading: Evidence from eye movements." In: *Quarterly Journal of Experimental Psychology* 61.5, pp. 708–723.

Just, Marcel A and Patricia A Carpenter (1980). "A theory of reading: From eye fixations to comprehension." In: *Psychological review* 87.4, pp. 329–354.

Kajiwara, Tomoyuki and Kazuhide Yamamoto (2015). "Evaluation dataset and system for Japanese lexical simplification." In: *Proceedings of the ACL-IJCNLP 2015 Student Research Workshop*, pp. 35–40.

Kalchbrenner, Nal and Phil Blunsom (2013). "Recurrent continuous translation models." In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1700–1709.

Katusic, Slavica K, Robert C Colligan, William J Barbaresi, Daniel J Schaid, and Steven J Jacobsen (2001). "Incidence of reading disability in a population-based birth cohort, 1976–1982, Rochester, Minn." In: *Mayo Clinic Proceedings*. Vol. 76. 11. Elsevier, pp. 1081–1092.

Kingma, Diederik P and Jimmy Ba (2014). "Adam: A method for stochastic optimization." In: *arXiv preprint arXiv:1412.6980*.

Klerke, Sigrid, Héctor Martínez Alonso, and Anders Søgaard (2015a). "Looking hard: Eye tracking for detecting grammaticality of automatically compressed sentences." In: *Proceedings of the 20th Nordic Conference of Computational Linguistics (NODALIDA 2015)*, pp. 97–105.

Klerke, Sigrid, Sheila Castilho, Maria Barrett, and Anders Søgaard (2015b). "Reading metrics for estimating task efficiency with MT output." In: *Proceedings of the Sixth Workshop on Cognitive Aspects of Computational Language Learning*, pp. 6–13.

Klerke, Sigrid, Yoav Goldberg, and Anders Søgaard (2016). "Improving sentence compression by learning to predict gaze." In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1528–1533.

Klerke, Sigrid, Janus Askø Madsen, Emil Juul Jacobsen, and John Paulin Hansen (2018). "Substantiating Reading Teachers with Scanpaths." In:

Knight, Kevin and Daniel Marcu (2000). "Statistics-based summarization-step one: Sentence compression." In: *AAAI/IAAI* 2000, pp. 703–710.

Knight, Kevin and Daniel Marcu (2002). "Summarization beyond sentence extraction: A probabilistic approach to sentence compression." In: *Artificial Intelligence* 139.1, pp. 91–107.

Koehn, Philipp (2009). *Statistical machine translation*. Cambridge University Press.

Koehn, Philipp et al. (2007). "MOSES: Open source Toolkit for Statistical Machine Translation." In: *Proceedings of ACL - Demonstration Session*, pp. 177–180. URL: http://dl.acm.org/citation.cfm?id=1557769.1557821.

Kučera, Henry and Winthrop Nelson Francis (1967). *Computational analysis of present-day American English*. Dartmouth Publishing.

Lal, Partha and Stefan Ruger (2002). "Extract-based summarization with simplification." In: *Proceedings of the ACL*.

Leacock, Claudia and Martin Chodorow (2003). "Automated grammatical error detection." In: *Automated essay scoring: A cross-disciplinary perspective*, pp. 195–207.

Lee, John and J. Buddhika K. Pathirage Don (2017). "Splitting Complex English Sentences." In: *Proceedings of the 15th International Conference on Parsing Technologies*. Pisa, Italy: Association for Computational Linguistics, pp. 50–55. URL: http://aclweb.org/anthology/W17-6307.

Lee, John and Chak Yan Yeung (2018). "Personalizing Lexical Simplification." In: *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*. Ed. by Emily M. Bender, Leon Derczynski, and Pierre Isabelle. Association for Computational Linguistics, pp. 224–232. ISBN: 978-1-948087-50-6. URL: https://aclanthology.info/papers/C18-1019/c18-1019.

Ligozat, Anne-Laure, Anne Garcia-Fernandez, Cyril Grouin, and Delphine Bernhard (2012). "Annlor: a naïve notation-system for lexical outputs ranking." In: *Proceedings of the 6th International Workshop on Semantic Evaluation*. Association for Computational Linguistics, pp. 487–492.

Liu, Jun and Yuji Matsumoto (2016). "Simplification of Example Sentences for Learners of Japanese Functional Expressions." In: *Proceedings of the 3rd Workshop on Natural Language Processing Techniques for Educational Applications (NLPTEA2016)*, pp. 1–5.

Long, Michael H and Steven Ross (1993). "Modifications That Preserve Language and Content." In: *Technical Report (ERIC)*.

Luong, Minh-Thang, Hieu Pham, and Christopher D Manning (2015). "Effective approaches to attention-based neural machine translation." In: *arXiv preprint arXiv:1508.04025*.

Luong, Minh-Thang, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser (2016). "Mutli-task sequence-to-sequence learning." In: *ICLR*.

Mandya, Angrosh A., Tadashi Nomoto, and Advaith Siddharthan (2014). "Lexico-syntactic text simplification and compression with typed dependencies." In: *COLING*, pp. 1996–2006.

Manning, Christopher D., Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky (2014). “The Stanford CoreNLP Natural Language Processing Toolkit.” In: *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 55–60. URL: http://www.aclweb.org/anthology/P/P14/P14-5010.

Marcus, Mitchell, Mary Marcinkiewicz, and Beatrice Santorini (1993). “Building a large annotated corpus of English: the Penn Treebank.” In: *Computational Linguistics* 19.2, pp. 313–330.

Martínez Alonso, Héctor and Barbara Plank (2017). “When is multitask learning effective? Semantic sequence prediction under varying data conditions.” In: *15th Conference of the European Chapter of the Association for Computational Linguistics*.

Mason, Jana M and Janet Ross Kendall (1979). “Facilitating Reading Comprehension through Text Structure Manipulation.” In: *Alberta Journal of Educational Research* 25.2, pp. 68–76.

McDonald, Ryan T (2006). “Discriminative Sentence Compression with Soft Syntactic Evidence.” In: *EACL*.

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean (2013). “Distributed representations of words and phrases and their compositionality.” In: *Advances in neural information processing systems*, pp. 3111–3119.

Miller, George (1998). *WordNet: An electronic lexical database*. MIT press.

Mou, Lili, Zhao Meng, Rui Yan, Ge Li, Yan Xu, Lu Zhang, and Zhi Jin (2016). “How Transferable are Neural Networks in NLP Applications?” In: *EMNLP*.

Mueller, Thomas, Helmut Schmid, and Hinrich Schütze (2013). “Efficient Higher-Order CRFs for Morphological Tagging.” In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Seattle, Washington, USA: Association for Computational Linguistics, pp. 322–332. URL: http://www.aclweb.org/anthology/D13-1032.

Napoles, Courtney, Chris Callison-Burch, Juri Ganitkevitch, and Benjamin Van Durme (2011). “Paraphrastic sentence compression with a character-based metric: Tightening without deletion.” In: *Proceedings of the Workshop on Monolingual Text-To-Text Generation*. Association for Computational Linguistics, pp. 84–90.

Narayan, Shashi and Claire Gardent (2014). “Hybrid Simplification using Deep Semantics and Machine Translation.” In: *Proceedings of ACL*, pp. 435–445. URL: http://www.aclweb.org/anthology/P14-1041.

Nisioi, Sergiu, Sanja Štajner, Simone Paolo Ponzetto, and Liviu P. Dinu (2017a). “Exploring Neural Text Simplification Models.” In: *Proceedings of ACL*, pp. 85–91. URL: http://aclweb.org/anthology/P17-2014.

Nisioi, Sergiu, Sanja Štajner, Simone Paolo Ponzetto, and Liviu P Dinu (2017b). "Exploring neural text simplification models." In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Vol. 2, pp. 85–91.

Nomoto, Tadashi (2007). "Discriminative sentence compression with conditional random fields." In: *Information Processing and Management: an International Journal* 43.6, pp. 1571–1587.

Nyström, Marcus and Kenneth Holmqvist (2010). "An adaptive algorithm for fixation, saccade, and glissade detection in eyetracking data." In: *Behavior research methods* 42.1, pp. 188–204.

O'Connor, Irene M. and Perry D. Klein (2004). "Exploration of strategies for facilitating the reading comprehension of high-functioning students with autism spectrum disorders." In: *Journal of autism and developmental disorders* 34.2, pp. 115–127.

Paetzold, Gustavo H. and Lucia Specia (2015). "Lexenstein: A framework for lexical simplification." In: *Proceedings of ACL-IJCNLP 2015 System Demonstrations*, pp. 85–90.

Paetzold, Gustavo H. and Lucia Specia (2016a). "Anita: An Intelligent Text Adaptation Tool." In: *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*, pp. 79–83.

Paetzold, Gustavo H. and Lucia Specia (2016b). "Semeval 2016 task 11: Complex word identification." In: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pp. 560–569.

Paetzold, Gustavo H. and Lucia Specia (2016c). "Sv000gg at semeval-2016 task 11: Heavy gauge complex word identification with system voting." In: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pp. 969–974.

Paetzold, Gustavo H. and Lucia Specia (2016d). "Unsupervised Lexical Simplification for Non-Native Speakers." In: *Proceedings of The 30th AAAI*, pp. 3761–3767.

Paetzold, Gustavo H. and Lucia Specia (2016e). "Vicinity-Driven Paragraph and Sentence Alignment for Comparable Corpora." In: *CoRR* abs/1612.04113. URL: http://arxiv.org/abs/1612.04113.

Paetzold, Gustavo H. and Lucia Specia (2017a). "A survey on lexical simplification." In: *Journal of Artificial Intelligence Research* 60, pp. 549–593.

Paetzold, Gustavo H. and Lucia Specia (2017b). "Lexical Simplification with Neural Ranking." In: *Proceedings of the 15th EACL*. Association for Computational Linguistics, pp. 34–40.

Paetzold, Gustavo H. and Lucia Specia (2017c). "Lexical Simplification with Neural Ranking." In: *Proceedings of EACL*, pp. 34–40. URL: http://www.aclweb.org/anthology/E17-2006.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu (2002). "BLEU: a method for automatic evaluation of machine translation." In: *Proceedings of ACL*, pp. 311–318. URL: http://www.aclweb.org/anthology/P02-1040.pdf.

Pedersen, Bolette, Sanni Nimb, Jørg Asmussen, Nicolai Sørensen, Lars Trap-Jensen, and Henrik Lorentzen (2009). "Dannet: the challenge of compiling a wordnet for danish by reusing a monolingual dictionary." In: *Language resources and evaluation* 43.3, pp. 269–299.

Petersen, Sarah E and Mari Ostendorf (2007). "Text simplification for language learners: a corpus analysis." In: *Workshop on Speech and Language Technology in Education*.

Peterson, Robin L and Bruce F Pennington (2015). "Developmental dyslexia." In: *Annual review of clinical psychology* 11, pp. 283–307.

Petrov, Slav, Dipanjan Das, and Ryan McDonald (2011). "A universal part-of-speech tagset." In: *arXiv preprint arXiv:1104.2086*.

Pitler, Emily (2010). *Methods for sentence compression*. Tech. rep. Department of Computer and Information Science, University of Pennsylvania. URL: http://repository.upenn.edu/cis_reports/929.

Plank, Barbara (2016). "Keystroke dynamics as signal for shallow syntactic parsing." In: *COLING*.

Porter, Martin F (2001). *Snowball: A language for stemming algorithms*.

Quigley, SP and PV Paul (1984). *Language and Deafness.*

Quinlan, Philip T (1992). *The Oxford psycholinguistic database*. University Press.

Rayner, Keith (1998). "Eye movements in reading and information processing: 20 years of research." In: *Psychological bulletin* 124.3, p. 372.

Rayner, Keith, Sara C Sereno, Robin K Morris, A Rene Schmauder, and Charles Clifton Jr (1989). "Eye movements and on-line language comprehension processes." In: *Language and Cognitive Processes* 4.3-4, SI21–SI49.

Rayner, Keith, Gary E Raney, and Alexander Pollatsek (1995). "Eye movements and discourse processing." In:

Rayner, Keith, Gretchen Kambe, and Susan A Duffy (2000). "The effect of clause wrap-up on eye movements during reading." In: *The Quarterly Journal of Experimental Psychology Section A* 53.4, pp. 1061–1080.

Rei, Marek (2017). "Semi-supervised Multitask Learning for Sequence Labeling." In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, pp. 2121–2130.

Rello, Luz, Ricardo Baeza-Yates, Laura Dempere-Marco, and Horacio Saggion (2013a). "Frequent words improve readability and short words improve understandability for people with dyslexia." In: *Human-Computer Interaction–INTERACT 2013*. Springer, pp. 203–219.

Rello, Luz, Ricardo Baeza-Yates, Stefan Bott, and Horacio Saggion (2013b). “Simplify or help?: text simplification strategies for people with dyslexia.” In: *Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility*. ACM, p. 15.

Riezler, Stefan, Tracy H King, Richard Crouch, and Annie Zaenen (2003). “Statistical sentence condensation using ambiguity packing and stochastic disambiguation methods for lexical-functional grammar.” In: *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*. Association for Computational Linguistics, pp. 118–125.

Ronzano, Francesco, Luis Espinosa Anke, Horacio Saggion, et al. (2016). “Taln at semeval-2016 task 11: Modelling complex words by contextual, lexical and semantic features.” In: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pp. 1011–1016.

Rudell, Alan P (1993). “Frequency of word usage and perceived word difficulty: Ratings of Kučera and Francis words.” In: *Behavior Research Methods, Instruments, & Computers* 25.4, pp. 455–463.

Sagot, Benoît and Darja Fišer (2008). “Building a free French wordnet from multilingual resources.” In: *OntoLex*.

Scarton, Carolina and Lucia Specia (2018). “Learning Simplifications for Specific Target Audiences.” In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Vol. 2, pp. 712–718.

Schneider, Nathan and Noah A Smith (2015). “A Corpus and Model Integrating Multiword Expressions and Supersenses.” In: *Proc. of NAACL-HLT. Denver, Colorado, USA.*

Sennrich, Rico, Barry Haddow, and Alexandra Birch (2016). “Edinburgh Neural Machine Translation Systems for WMT 16.” In: *Proceedings of WMT*, pp. 371–376. URL: http://www.statmt.org/wmt16/pdf/W16-2323.pdf.

Sennrich, Rico et al. (2017). “Nematus: a Toolkit for Neural Machine Translation.” In: *Proceedings of the Software Demonstrations of EACL*. Valencia, Spain: ACL, pp. 65–68. URL: http://aclweb.org/anthology/E17-3017.

Shardlow, Matthew (2013a). “A Comparison of Techniques to Automatically Identify Complex Words.” In: *51st Annual Meeting of the Association for Computational Linguistics Proceedings of the Student Research Workshop*, pp. 103–109.

Shardlow, Matthew (2013b). “The cw corpus: A new resource for evaluating the identification of complex words.” In: *Proceedings of the Second Workshop on Predicting and Improving Text Readability for Target Reader Populations*, pp. 69–77.

Shardlow, Matthew (2014a). “A survey of automated text simplification.” In: *International Journal of Advanced Computer Science and Applications* 4.1, pp. 58–70.

Shardlow, Matthew (2014b). “Out in the Open: Finding and Categorising Errors in the Lexical Simplification Pipeline.” In: *LREC*, pp. 1583–1590.

Shatz, Itamar (2017). “Native Language Influence During Second Language Acquisition: A Large-Scale Learner Corpus Analysis.” In: *Proceedings of the Pacific Second Language Research Forum (PacSLRF 2016)*, pp. 175–180.

Shaywitz, Sally E, Michael D Escobar, Bennett A Shaywitz, Jack M Fletcher, and Robert Makuch (1992). “Evidence that dyslexia may represent the lower tail of a normal distribution of reading ability.” In: *New England Journal of Medicine* 326.3, pp. 145–150.

Shravan, Vasishth, Brüssow Sven, Lewis Richard L., and Drenhaus Heiner (2010). “Processing Polarity: How the Ungrammatical Intrudes on the Grammatical.” In: *Cognitive Science* 32.4, pp. 685–712. DOI: 10.1080/03640210802066865. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1080/03640210802066865. URL: https://onlinelibrary.wiley.com/doi/abs/10.1080/03640210802066865.

Siddharthan, Advaith (2003). “Resolving pronouns robustly: Plumbing the depths of shallowness.” In: *Proceedings of the 2003 EACL Workshop on The Computational Treatment of Anaphora*.

Siddharthan, Advaith (2006). “Syntactic simplification and text cohesion.” In: *Research on Language and Computation* 4.1, pp. 77–109.

Siddharthan, Advaith (2014). “A survey of research on text simplification.” In: *ITL-International Journal of Applied Linguistics* 165.2, pp. 259–298.

Siddharthan, Advaith and Angrosh Annayappan Mandya (2014). “Hybrid text simplification using synchronous dependency grammars with hand-written and automatically harvested rules.” In: *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2014)*. Association for Computational Linguistics.

Sinha, Ravi (2012). “Unt-simprank: Systems for lexical simplification ranking.” In: *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*. Association for Computational Linguistics, pp. 493–496.

Smith, Jason R, Chris Quirk, and Kristina Toutanova (2010). “Extracting parallel sentences from comparable corpora using document level alignment.” In: *Proceedings of NAACL*, pp. 403–411. URL: http://www.aclweb.org/anthology/N10-1063.

Snover, Matthew, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul (2006). “A study of translation edit rate with targeted human annotation." In: *Proceedings of AMTA*, pp. 223–231. URL: https://www.cs.umd.edu/~snover/pub/amta06/ter_amta.pdf.

Søgaard, Anders and Yoav Goldberg (2016). "Deep multitask learning with low level tasks supervised at lower layers." In: *ACL*.

Specia, Lucia (2010). "Translating from complex to simplified sentences." In: *International Conference on Computational Processing of the Portuguese Language*. Springer, pp. 30–39.

Specia, Lucia, Sujay Kumar Jauhar, and Rada Mihalcea (2012). "SemEval-2012 task 1: English lexical simplification." In: *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*. Association for Computational Linguistics, pp. 347–355.

Spitkovsky, Valentin I, Daniel Jurafsky, and Hiyan Alshawi (2010). "Profiting from mark-up: Hyper-text annotations for guided parsing." In: *ACL*.

Stanojević, Maja (2009). "Cognitive synonymy: A general overview." In: *Facta universitatis-series: Linguistics and Literature* 7.2, pp. 193–200.

Sultan, Md, Steven Bethard, and Tamara Sumner (2014). "Back to Basics for Monolingual Alignment: Exploiting Word Similarity and Contextual Evidence." In: *TACL* 2, pp. 219–230. ISSN: 2307-387X. URL: http://www.aclweb.org/anthology/Q14-1018.

Sutskever, Ilya, Oriol Vinyals, and Quoc V Le (2014). "Sequence to sequence learning with neural networks." In: *Advances in neural information processing systems*, pp. 3104–3112.

Tiedemann, Jörg, Željko Agić, and Joakim Nivre (2014). "Treebank Translation for Cross-Lingual Parser Induction." In: *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*. Ann Arbor, Michigan: Association for Computational Linguistics, pp. 130–140.

To, Hoai-Viet, Ryutaro Ichise, and Hoai-Bac Le (2009). "An adaptive machine learning framework with user interaction for ontology matching." In: *Proceedings of the International Joint Conferences on Artifical Intelligence, Workshop on Information Integration on the Web*, pp. 35–40.

Traxler, Matthew J (2002). "Plausibility and subcategorization preference in children's processing of temporarily ambiguous sentences: Evidence from self-paced reading." In: *The Quarterly Journal of Experimental Psychology: Section A* 55.1, pp. 75–96.

Turner, Jenine and Eugene Charniak (2005). "Supervised and unsupervised learning for sentence compression." In: *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, pp. 290–297.

Tweissi, Adel I (1998). "The Effects of the Amount and Type of Simplification on Foreign Language Reading Comprehension." In: *Reading in a foreign language* 11.2, pp. 191–204.

Watanabe, Willian Massami, Arnaldo Candido Junior, Vinícius Rodriguez Uzêda, Renata Pontin de Mattos Fortes, Thiago Alexandre Salgueiro Pardo, and Sandra Maria Aluísio (2009). "Facilita: reading assistance for low-literacy readers." In: *Proceedings of the 27th ACM international conference on Design of communication*. ACM, pp. 29–36.

Watson, Betty U and David E Goldgar (1988). "Evaluation of a typology of reading disability." In: *Journal of clinical and experimental neuropsychology* 10.4, pp. 432–450.

Woodsend, Kristian and Mirella Lapata (2011). "Learning to simplify sentences with quasi-synchronous grammar and integer programming." In: *Proceedings of the conference on empirical methods in natural language processing*. Association for Computational Linguistics, pp. 409–420.

Wubben, Sander, Antal Van Den Bosch, and Emiel Krahmer (2012). "Sentence simplification by monolingual machine translation." In: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*. Association for Computational Linguistics, pp. 1015–1024.

Xu, Wei, Chris Callison-Burch, and Courtney Napoles (2015). "Problems in current text simplification research: New data can help." In: *TACL* 3, pp. 283–297. URL: https://transacl.org/ojs/index.php/tacl/article/view/549/131.

Xu, Wei, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch (2016). "Optimizing statistical machine translation for text simplification." In: *Transactions of the Association for Computational Linguistics* 4, pp. 401–415.

Yaneva, Victoria (2016). "Assessing text and web accessibility for people with autism spectrum disorder." PhD thesis. University of Wolverhampton.

Yaneva, Victoria and Richard Evans (2015). "Six good predictors of autistic text comprehension." In: *Proceedings of the International Conference Recent Advances in Natural Language Processing*, pp. 697–706.

Yaneva, Victoria, Irina P Temnikova, and Ruslan Mitkov (2016a). "Evaluating the Readability of Text Simplification Output for Readers with Cognitive Disabilities." In: *LREC*.

Yaneva, Victoria, Richard Evans, and Irina Temnikova (2016b). "Predicting Reading Difficulty for Readers with Autism Spectrum Disorder." In: *Proceedings of Workshop on Improving Social Inclusion using NLP: Tools and Resources (ISI-NLP) held in conjunction with LREC 2016*. Portoroz, Slovenia.

Yatskar, Mark, Bo Pang, Cristian Danescu-Niculescu-Mizil, and Lillian Lee (2010). "For the sake of simplicity: Unsupervised extraction of lexical simplifications from Wikipedia." In: *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, pp. 365–368.

Yimam, Seid Muhie and Chris Biemann (2018). "Par4Sim - Adaptive Paraphrasing for Text Simplification." In: *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*. Ed. by Emily M. Bender, Leon Derczynski, and Pierre Isabelle. Association for Computational Linguistics, pp. 331–342. ISBN: 978-1-948087-50-6. URL: https://aclanthology.info/papers/C18-1028/c18-1028.

Yimam, Seid Muhie, Sanja Štajner, Martin Riedl, and Chris Biemann (2017). "Multilingual and Cross-Lingual Complex Word Identification." In: *Proceedings of RANLP*, pp. 813–822.

Yimam, Seid Muhie, Chris Biemann, Shervin Malmasi, Gustavo H. Paetzold, Lucia Specia, Sanja Štajner, Anaïs Tack, and Marcos Zampieri (2018). "A Report on the Complex Word Identification Shared Task 2018." In: *Proceedings of the 13th Workshop on Innovative Use of NLP for Building Educational Applications*. New Orleans, United States: Association for Computational Linguistics.

Zhang, Xingxing and Mirella Lapata (2017). "Sentence Simplification with Deep Reinforcement Learning." In: *Proceedings of EMNLP*, pp. 595–605. URL: https://www.aclweb.org/anthology/D17-1063.

Zhang, Yaoyuan, Zhenxu Ye, Yansong Feng, Dongyan Zhao, and Rui Yan (2017). "A Constrained Sequence-to-Sequence Neural Model for Sentence Simplification." In: *CoRR* abs/1704.02312. URL: https://arxiv.org/abs/1704.02312.

Zhu, Zhemin, Delphine Bernhard, and Iryna Gurevych (2010). "A monolingual tree-based translation model for sentence simplification." In: *Proceedings of the 23rd international conference on computational linguistics*. Association for Computational Linguistics, pp. 1353–1361.

Ziegler, Johannes C, Caroline Castel, Catherine Pech-Georgel, Florence George, F-Xavier Alario, and Conrad Perry (2008). "Developmental dyslexia and the dual route model of reading: Simulating individual differences and subtypes." In: *Cognition* 107.1, pp. 151–178.

Žliobaitė, Indrė, Mykola Pechenizkiy, and Joao Gama (2016). "An overview of concept drift applications." In: *Big data analysis: new algorithms for a new society*. Springer, pp. 91–114.

COLOPHON

This document was typeset using the typographical look-and-feel `classicthesis` developed by André Miede and Ivo Pletikosić. The style was inspired by Robert Bringhurst's seminal book on typography *"The Elements of Typographic Style"*. Joachim Bingel adapted the style for this thesis.

Final Version as of August 31, 2018 (`classicthesis` version 1.0).