DeScipher: A Text Simplification Tool for Science Journalism

Yea Seul Kim
University of Washington
1410 NE Campus Parkway
Seattle, WA 98195
yeaseul1@uw.edu

Jessica Hullman
University of Washington
1410 NE Campus Parkway
Seattle, WA 98195
jhullman@uw.edu

Eytan Adar
University of Michigan
500 South State Street
Ann Arbor, MI 48109
eadar@umich.edu

ABSTRACT

Complex jargon often makes scientific work less accessible to the general public. By employing a set of specific reporting strategies, journalists bridge these groups by delivering information about scientific advances in a readable, engaging way. One such strategy is using simpler terms in place of complex jargon. To assist in this process, we introduce DeScipher, a text editor application that suggests and ranks possible simplifications of complex terminology to a journalist while she is authoring an article. DeScipher applies simplification rules derived from a large collection of scientific abstracts and associated author summaries, and accounts for textual context in making suggestions to the journalist. In evaluating our system, we show that DeScipher is a viable application for producing useful simplifications of scientific and other terms by comparing to prior techniques used on other corpora. We also propose concrete opportunities for future development of "journalist-in-the-loop" tools for aiding journalists in enacting science reporting strategies.

Keywords

Science reporting, Text simplification, Human-in-the-loop

1. INTRODUCTION

Journalists play an important role in translating scientific news for public consumption. For example, a journalist reporting on a new nanoscience technology might make the information more accessible by describing how the size or features of the device using terms that are more familiar to her audience. To broker between scientists and the public requires the journalist to think about the science from multiple perspectives. She must come to understand the scientific contribution (e.g., a new technology that can produce a pin-sized computer chip) without overlooking important details (e.g., the possibly subtle differences from the previous state-of-the-art solution or production details). Often this requires the journalist to make sense of the reported findings in publications intended for domain scientists. At the same time, she must leverage her understanding of her audience to reformulate the novel scientific information in terms they will understand.

We propose that computer scientists work with journalists to explore the design space of "journalist-in-the-loop" tools for science reporting: systems that leverage automated methods to make the journalist's process more efficient. In particular, we suggest that scientific text corpora and other large datasets can be combined with natural language processing approaches to re-express, summarize, and structure scientific information for presentation to the public.

To identify concrete opportunities for such systems, we consulted guidelines for science reporting (where the journalist is doing the 'translation') as well as advice for scientists speaking with journalists (where the scientists themselves are translating). Many suggestions focus on the need to simplify scientific content. Journalists and scientists alike are advised to avoid scientific jargon and abbreviations [5, 11]: e.g,. use 'sodium' instead of 'Na', 'milligrams' instead of 'mg', and 'antibiotic' rather than 'cephalosporin.' Automated approaches to these types of simplifications could help make the journalist's process more efficient whenever scientific content needs to be expressed (e.g., a science-themed article or general news with a scientific component, like an earthquake). A system that identifies and proposes viable simplifications could save valuable time that would otherwise be spent searching for suitable words.

We present DeScipher, a system to help journalists or editors in automatically suggesting simplifications for scientific terms and other complex jargon. The tool can be applied both when reading old material or writing new text. DeScipher applies context-aware lexical simplifications derived from a corpus of over 15,000 abstracts and simpler author summaries for articles in *Public Library Of Science* (PLOS) journals. Our work is the first to apply lexical simplification suggestions in a tool for science journalists. We describe the user experience and text simplification pipeline for DeScipher. We present example simplifications and evaluate these results against prior work in lexical simplification rules learned from Wikipedia.

We reflect on opportunities to extend DeScipher's capabilities for science reporting and potential uses in other complex reporting domains (e.g., political, financial, etc.). We conclude by proposing a concrete set of opportunities for future development of interfaces that leverage automation to facilitate other strategies used by journalists.

2. BACKGROUND

We describe previous work for journalist support as well as text simplification more broadly.

2.1 Automated Support for Journalists' Work

Journalists and researchers are developing increasingly sophisticated methods for making the news generation pipeline more efficient. These include tools for fact checking (e.g., [23]), generating content[1], and creating graphics (e.g., [10]),

[1]http://www.narrativescience.com/,

to name a few. While sophisticated thesauri can be used to help with simplification, these focus on broad classes of synonyms and not on simplification (where other categories, such as hypernyms, are useful). We are inspired by these systems in our exploration of applying NLP approaches to support another task in the news creation process: simplification and re-expression of scientific information.

2.2 Text Simplification

Various text simplification methods have been developed to reduce syntactic or lexical complexity in a piece of text without distorting the meaning (see, e.g., [6]). Our work focuses on *lexical simplification*, the process of reducing the use of highly complex words in a piece of text. A common approach for this kind of simplification is to: (1) identify pairs of words with certain semantic relationships (for example synonyms like *saccharides* and *sugars* or hyponym-hypernym pairs like *lepidopterans* and *insects*), and then (2) predict if the candidate replacement is in fact simpler using various features (e.g., length of each word in characters [2, 18] or the ratio of occurrences of words in corpora known to be complex versus simple [2, 22]). An input text can then be simplified by substituting occurrences of more complex words with their simpler synonym or hypernym. Research has targeted each of these steps individually and together as part of a pipeline.

A classic method for finding viable synonym and hyponym-hypernym pairs is to look for the occurrence of simple patterns like "an A, such as B" in the corpus [8]. Because only a small set of patterns are used, this approach may not work with smaller corpora (the likelihood of observing a pair of terms that matches the template is small). An alternative method, which we adapt, is to identify all content words in a corpus (excluding stop words like "the" and other special tokens), and then to consider as a candidate pair any two content words within a given proximity window (e.g., within 10 tokens) [2]. Though this is a rough proxy, pairs can be further filtered to improve accuracy (e.g., by considering contexts for both the existing and potentially simpler terms in the broader corpora [2, 9]). For DeScipher we adapt the technique proposed by Biran et al. [2].

Most work in lexical simplification has focused on a somewhat narrow range of related corpora (where most often one of them is Wikipedia): user-generated health forum content versus Wikipedia [22], or Simple English Wikipedia versus English Wikipedia [2, 9, 24]. Rather than utilize Wikipedia we leverage a fairly recent set of requirements by journals (PLOS and PNAS) and grant agencies (e.g., NSF): authors must now create a parallel abstract for their work that is intended for consumption by the general public.

We believe that this pairing of corpora has benefits over other training sets. First, the two abstracts are naturally 'parallel,' and both are authored by the scientists themselves. Second, terms associated with new scientific advances may not yet be reflected in corpora such as Wikipedia. Third, Wikipedia's editing rules favor generality over specificity and eliminate redundancy as much as possible (both are problematic in providing a rich training set). Finally, a corpus of scientific abstracts and summaries (rather than a broad article focused on a specific 'topic') may provide more content around any given scientific topic, in turn leading to more simplifications for science journalism.

http://automatedinsights.com/

To our knowledge our work is the first to propose developing text simplification tools expressly for journalists as part of the text editor. While the idea of developing text editors that could aid writers by identifying places to improve a text dates to systems proposed in 1980's [14, 12], few existing text editors support simplification of text in context. While not a text editor, Aluisio et al. [1] describe ProSimple, a reader-driven system that allows readers of the Brazilian Portugeuse text dataset to apply various text simplifications. Journalists might benefit from ProSimple's broad notion of simplification. However, we believe that by focusing on scientific language we can better address the concerns of scientific reporting.

3. THE DESCIPHER SYSTEM

DeScipher is a text-editor application that helps journalists identify and re-express scientific and other terms in simpler ways via ranked, context-aware suggestions. We describe the user experience and system architecture.

3.1 User Experience

Imagine a journalist is taking notes in preparation for writing a news article as she reads about a topic in scientific articles. She pastes useful content for her article into DeScipher, and the system automatically underlines complex terminology that could be expressed more simply. When the journalist selects an underlined term that she is not familiar with (e.g., *cytokines*, the tool shows a list of simpler words with a similar meaning (e.g., *protein*) (Figure 1, left).

The same interactive editor can be used as the journalist transitions to drafting the article text. For each sentence she enters, DeScipher automatically underlines complex words and suggests simpler words when she selects one of the underlined words (Figure 1, right). If she decides to use one of the simplifications in the list, she can click on it and the word in the passage is replaced by the simpler term, or she can add the simpler term in parentheses after the complex term (e.g., *protein*). DeScipher also displays information about which complex words are best to replace with simplifications by adding one or more '*' beside the terms. We envision extending DeScipher to include an automated readability assessment function that allows an author to check the reading level of an article draft, and suggests term simplifications that could help her achieve a target reading level.

3.2 System Architecture

3.2.1 Data

We extracted 15,867 of abstracts and author summaries from various PLOS journals created between 2010 and 2014; 3,681 from PLOS Pathogens, 680 from PLOS Medicine, 1,384 from PLOS Biology, 3,064 from PLOS Computational Biology, 4,284 from PLOS Genetics and 2,774 of PLOS Neglected Tropical Diseases. PLOS publications provide guidance to authors on how to differentiate these two descriptions. In the author summary section, authors are guided to use simpler and non-technical terms [7]. The main purpose of this section is to make the article more accessible to a wider (non-scientific) audience. In contrast, the abstract is intended to describe the contributions succinctly and clearly for readers without necessarily simplifying the language. [2]

[2]journals.plos.org/plosgenetics/s/submission-guidelines

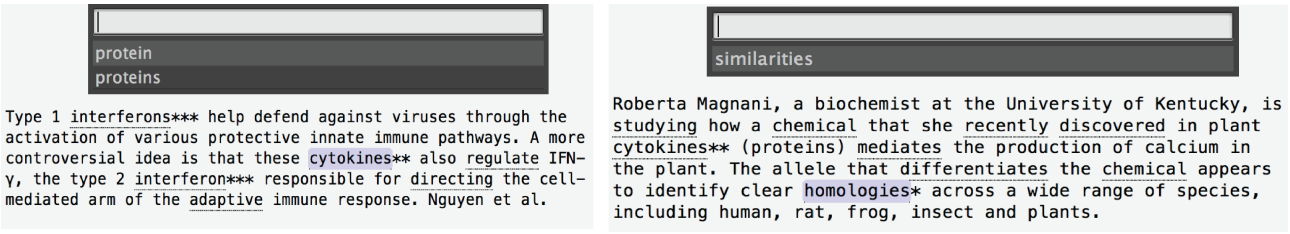

Figure 1: DeScipher provides simplifications to a journalist as she *reads* scientific articles on a topic (left) or suggestions as she *drafts* her news article (right). Annotations (labeled with '*') signal which simplifications are most likely to be helpful (the more *'s, the more likely simplification will help).

3.2.2 *Text Simplification Pipeline*

We adapt Biran et al.'s context aware lexical simplification technique [2] which generates simplification rules given a corpus divided into complex and simple documents (i.e., "standard" and "Simple English" versions of Wikipedia in the original implementation). Each rule consists of a pair of words where the second word can be used to simplify the first (e.g., *acaricides* and *pesticides*) along with a score indicating the similarity of the two words based on the other words they co-occur with. While Biran et al. use the simpler corpus only to estimate the frequency of terms in non-expert usage (as do [9, 22]), we extract simplification rules from the author summaries in addition to the abstracts.

Identifying Simplifications

The first stage of the pipeline consists of identifying simplification rules. We first find *content words* in the combined abstracts and author summaries (i.e., words that remain after eliminating stop words, numbers and punctuation). For all possible pairs of content words we observe, we filter using the following: stem both words (using the Porter stemmer provided by the Python NLTK library [3]) and omit those pairs which share a lemma (e.g., *permutable*, *permutation*); tag the part of speech (POS) of each word using Morphadorner [4] and omit those for which the POS differs (e.g., *permutation* (noun), *change;d* (verb)); check that the pairs have a synonym or hypernym relation to each other using WordNet [15], and exclude those that do not.

After the above filtering, 91,553 pairs remain. We then ensure that one word is, in fact, simpler (our goal is not to replace complexity with complexity). To do so, we first calculate the corpus complexity of each word w in a pair as the ratio between the frequency of occurrence of w in the complex versus simple corpus:

$$C_w = \left(\frac{f_{w,abstract}}{f_{w,summary}}\right) \quad (1)$$

and the lexical complexity of w as $L_w = |w|$ (the length of the word). The final complexity for the word is:

$$X_w = (C_w \times L_w) \quad (2)$$

The more complex word in the pair is the word for which X_w is greater.

To ensure that the suggestions that are made by DeScipher are grammatically correct, we produce additional pairs for morphological variants of the original pair using Python Nodebox [3] (e.g., by generating other possible conjugations of verbs and other possible tenses for nouns). This process results in 28,841 total pairs.

Finally, for each unique content word w in the corpus, we create a context vector CV_w: a vector that records the frequency with which all other content words in the corpus appeared in the same sentence. This vector is later used to decide whether a candidate word for simplification should be replaced given the sentence it appears in.

Applying Simplifications

The goal of the second stage of the pipeline is to identify target words in an input text for simplification and to identify which simplification rules to apply. Following Biran et al. [2], we do not attempt simplification if a sentence has less than 7 tokens. For all other sentences we calculate the cosine similarity between the context vector CV_w of the target word w and a context vector for the sentence (s) $SCV_{s,w}$. $SCV_{s,w}$ is a vector of the frequency of occurrence of all unique content word in the sentence. If the cosine similarity is too high, the target word w may have been used for its precise meaning, and simplification may not be useful. For example, in the sentence "Accham are less than 1m tall at the withers and typically used for dairy rather than meat.", we would not want to replace "Accham" with "oxen" or "cattle" since this loses the original meaning of the sentence. While Biran et al. apply a threshold of 0.1, we found through experimentation that a slightly higher threshold of 0.4 can still produce viable simplification suggestions (e.g., *kinase* → *enzyme*, cos: 0.376 in the sentence "*In the Drosophila eye, cross-repression between the Hippo pathway kinase LATS/Warts and the growth regulator generates mutually exclusive types of photoreceptors*").

To ensure that the target word uses the same sense of the word that was seen in the corpus, Biran et al. also use a threshold on a second score of the context similarity between $SCV_{s,w}$ and a vector containing the minimum co-occurrence count of each word in the context vectors for the two words in the pair: $(CCV_{w,x}[i] = \min(CCV_{w,x}, SCV_{s,w}))$:

$$ContextSim = \cos(CCV_{w,x}, SCV_{s,w}) \quad (3)$$

However, this score's reliability in reflecting the quality of the sense match varies with the size of the corpus. Because we use a smaller corpus, and a journalist will ultimately decide which simplifications to use or to reject, we do not filter using this score but do use it to rank simplification candidates in the application (see 3.2.3).

Our method followed Biran et al's technique for checking that the complexity of the second word in each pair is less than that of the first word. However, because the complexity of a word w (eq. 2) considers both word length and ratio of occurrence in the complex to simple corpus, it is possi-

[3]https://www.nodebox.net/code/index.php/Web

ble that pairs are found in which the first (more complex) word in the pair appears equally or even more frequently in the simple corpus. To ensure that our application does not suggest simplifications in such cases, we use a threshold that requires the ratio between the complex and simple corpus frequency of the first (complex) word in the pair to be greater than 1.25.

3.2.3 User Interface

We implemented the user interface for DeScipher in the text editor Sublime text3[4]. Once text is entered into the editor, DeScipher identifies all words in the text for which simplifications are available. These terms are underlined to bring them to the journalist's attention. An underlined term can be selected by highlighting the text and using Command + t on Mac or Ctrl + t on Windows. We also convey to the journalist the quality of the best simplification available for a term by adding 1 or more '*' symbols after the term. The number of '*' are determined using the *ContextSim* score which provides some information on how well the current context of the target word matches that of the simplification rule (eq. 3). Specifically, the following ranges for *ContextSim* determine the '*' annotation: 0.02 to 0.05 → '*', 0.05 to 0.1 → '**', ≥ 0.1 → '***'.

4. EVALUATION

In total, our system produced 28,841 viable pairs from analyzing the corpus. To evaluate our pipeline, we applied our simplification rules to a new sample of 280 abstracts from research articles in Science from 2013 to 2015. We also applied Biran et al's [2] implementation[5] to the same sample and compared results.

DeScipher produced 457 unique simplification suggestions on the sample. Suggestions ranged from domain-specific scientific terminology simplifications (e.g., *murine* → *rodent*), to non-domain specific terms that are often used in scientific writing (e.g., *helices* → *curves*), to terms that are not necessarily scientific but could be simplified (e.g., *elucidation* → *clarification*). Biran et al.'s [2] pipeline produced 132 unique simplification pairs on the sample. While it is not surprising that their procedure produced less simplifications based on the more conservative thresholds, we observed that many of the simplifications that were produced involve two relatively familiar terms, making them less likely to help in re-expressing scientific content: e.g., *combat→fight*, *culture→society*, *maintain→hold*.

We presented 12 crowdworkers with each of the pairs from both methods, and asked them to confirm which of the two words was more familiar, and how well the more familiar term helped them understand the meaning of the complex term. While workers were slightly less likely to agree that the second word was simpler than the first in pairs produced on PLOS versus Wikipedia (55.5% vs 62.8%, $t(2442)$=5.05, $p < 0.001$), they were more likely to find our simplifications helpful in cases where the complex word was correctly identified ($f(1, 2817)$=0.76, $p < 0.001$). We believe that with a larger corpus, we can more reliably predict which word is simpler.

Table 1 shows a set of simplification pairs representing domain-specific or domain-general terms associated with science that were identified by DeScipher but were not identified by Biran et al.'s implementation.

Table 1: Examples of simplification pairs extracted from PLOS author summary and abstract corpus. Parentheses indicate the frequency in the complex and simple corpus, respectively.

Complex term	Simple term
progenitor (22,8)	ancestor (14,21)
inoculation (9,40)	vaccination (87,100)
candida (10,5)	fungus (10,23)
genomics (38,21)	genetics (60,56)
perturbations (23,15)	disturbances (2,3)
stratification (16,2)	classification (33,22)
proteome (6,4)	protein (654,722)
motility (38,27)	movement (18,30)
crystal (9,7)	solid (18,16)
stoichiometry (7,3)	ratio (7,3)
cytokines (23,15)	proteins (547,606)
Na (14,4)	sodium (3,10)
biosphere (3, 1)	region (125,134)

We have also applied DeScipher to existing articles about scientific topics from large and small news organizations. We observe that even for "finished" articles DeScipher can suggest useful simplifications that may aid readers.

5. DISCUSSION

5.1 Limitations and Extensions

We used a simplification technique that focuses on unigrams, though many scientific terms may be bigrams or trigrams (e.g., *bordetella pertussis* → *whooping cough*).

We plan to extend DeScipher in several other ways to improve its suggestions. One way is to increase the size of the corpus by adding additional corpora with complex and simpler descriptions of scientific work, such as the abstracts and significance statements in *PNAS* and abstracts and editor's summary in *Science*. We also believe that mining scientific publications plus news releases and existing news articles on a topic could be fruitful.

Incorporating a reading level assessment function could also increase the usefulness of our simplifications. We envision a function that calculates the reading level of the journalist's drafted article on demand using statistical models trained on the PLOS corpus (similar to [21]), and suggests simplifications that target specific reading levels.

One potential non-scientific use for DeScipher is to allow a news organization to add to the set of rules (pairs) to improve consistency in terminology across news articles. For example, an organization might prefer that all foreign affairs journalists refer to a foreign government using the same phrasing (e.g., *ISIS* instead of *IS*, *SIC*, *Da'ish*). We are interested in working with journalists to discover other opportunities to make DeScipher useful to reporters, copy-editors, and others in the news production pipeline.

DeScipher's ability to provide multiple re-expressions of complex terminology through interaction could also be useful to the news reader. Implementing DeScipher as a reading tool on a news site could help readers of varying literacy levels get expressions that are familiar to them.

[4]http://www.sublimetext.com/3

[5]http://www.cs.columbia.edu/~orb/code_data.html

5.2 Opportunities for Future Work

To identify opportunities for future development of tools, we surveyed manuals for journalists [11] as well as advice given to scientists on how to present their work in ways that facilitates engaging reporting [5, 16]. We describe several themes that emerged from our analysis and propose a particular instantiation that leverages automated techniques.

Other strategies related to **Simplification** include using short sentences and avoiding connective works [5, 11]. Techniques for syntactic simplification could be used in a system like DeScipher to further reduce the time the journalist spends revising and rewriting article drafts.

Concretization strategies aim to make unfamiliar scientific units and measurements (e.g., a Newton, 0.01 inches) more understandable. This can occur through analogies (e.g., "Scientists in China have invented a sewing thread so strong that it could take the weight of a fully-grown elephant" [11]). Such analogies could be generated using online databases of objects or landmarks like travel websites along with automated algorithms. Other concretization strategies include giving examples [5]. We envision adapting text mining to enable a journalist to quickly curate possible examples related to a method or topic from a scientific corpus like PLOS, using common phrase patterns (e.g., "For example, ", "For instance, ").

Contextualization strategies include providing background details on scientific advances so that news readers can understand the history and significance of a new discovery [11]. In the PLOS corpus, we noticed that background details contextualizing an advance often were given in the first few sentences of both abstracts and author summaries, and many scientific articles include background information in the introduction and related work sections. Applying text summarization techniques (e.g., [17, 19, 20]) targeted to such sections of scientific publications could make it easier for a journalist to quickly identify important context and express it in an article.

Finally, several strategies for news reporting concern **Structuring Content**. For example, scientists are advised to provide a short summary, then elaborate on a discovery, then present a statement describing the "So what" of the work in talking to journalists. We suspect that patterns may be identifiable using text mining techniques based on phrase structures, as well as the progression of the complexity of the writing through the article. An application might learn successful complexity progressions by analyzing the complexity of terminology and syntax in a corpus of award winning articles [13]. These patterns could be automatically compared to a journalist's draft in an evaluative feature similar to the reading level assessment function we propose above.

6. CONCLUSION

We presented DeScipher, a system for suggesting text simplifications to a science journalist as she parses scientific articles or drafts her own news report on a topic. We use DeScipher to demonstrate the potential for 'journalist-in-the-loop' tools to aid science reporters.

7. ACKNOWLEDGMENTS

We thank Matthew Burgess for contributing scraping tools.

8. REFERENCES

[1] S. M. Aluísio, L. Specia, T. A. Pardo, E. G. Maziero, and R. P. Fortes. Towards brazilian portuguese automatic text simplification systems. In *Proc. of Symposium on Document Engineering*.

[2] O. Biran, S. Brody, and N. Elhadad. Putting it simply: A context-aware approach to lexical simplification. In *ACL '11*, 2011.

[3] S. Bird, E. Klein, and E. Loper. *Natural language processing with Python.* " O'Reilly Media, Inc.", 2009.

[4] P. R. Burns. Morphadorner v2: a java library for the morphological adornment of english language texts. *Northwestern University, Evanston, IL*, 2013.

[5] A. Collings. Media training notes, 2003.

[6] L. Feng. Text simplification: A survey. *The City University of New York, Tech. Rep*, 2008.

[7] P. Genetics. Submission guidelines.

[8] M. A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *COLING '92*, 1992.

[9] C. Horn, C. Manduca, and D. Kauchak. In *ACL '14*.

[10] D. N. Hullman, Jessica and E. Adar. Contextifier: Automated generation of annotated stock visualizations. In *CHI '13*, 2013.

[11] D. Ingram and P. Henshall. The news manual online.

[12] R. T. Kellogg. Computer aids that writers need. *Behavior Research Methods, Instruments, & Computers*, 17(2).

[13] A. Louis and A. Nenkova. A corpus of science journalism for analyzing writing quality.

[14] N. H. Macdonald, L. T. Frase, P. S. Gingrich, and S. A. Keenan. The writer's workbench: Computer aids for text analysis. *Educational psychologist*, 17(3).

[15] G. A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, Nov. 1995.

[16] R. Olson. *Don't Be Such a Scientist: Talking Substance in an Age of Style*. Island Press, 2009.

[17] C. D. Paice and P. A. Jones. The identification of important concepts in highly structured technical papers. In *SIGIR '93*.

[18] L. Rello, R. Baeza-Yates, L. Dempere-Marco, and H. Saggion. Frequent words improve readability and short words improve understandability for people with dyslexia. In *INTERACT '12*.

[19] H. Saggion and G. Lapalme. Concept identification and presentation in the context of technical text summarization. In *Proc. of NAACL-ANLP '00*.

[20] H. Saggion and G. Lapalme. Where does information come from? corpus analysis for automatic abstracting, 1998.

[21] L. Si and J. Callan. A statistical model for scientific readability. In *Proc. of CIKM '01*, 2001.

[22] V. Vydiswaran, Q. Mei, D. A. Hanauer, and K. Zheng. Mining consumer health vocabulary from community-generated text. In *AMIA '14*, 2014.

[23] B. Walenz and e. al. Finding, monitoring, and checking claims computationally based on structured data.

[24] M. Yatskar, B. Pang, C. Danescu-Niculescu-Mizil, and L. Lee. For the sake of simplicity: Unsupervised extraction of lexical simplifications from wikipedia. In *NAACL '10*, 2010.