Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 5721–5729
Marseille, 11–16 May 2020
© European Language Resources Association (ELRA), licensed under CC-BY-NC

wikiHowToImprove: A Resource and Analyses on Edits in Instructional Texts

Talita Rani Anthonio*, Irshad Ahmad Bhat*, Michael Roth
University of Stuttgart
Institute for Natural Language Processing
{anthonta,bhatid,rothml}@ims.uni-stuttgart.de

Abstract

Instructional texts, such as articles in wikiHow, describe the actions necessary to accomplish a certain goal. In wikiHow and other resources, such instructions are subject to revision edits on a regular basis. Do these edits improve instructions only in terms of style and correctness, or do they provide clarifications necessary to follow the instructions and to accomplish the goal? We describe a resource and first studies towards answering this question. Specifically, we create wikiHowToImprove, a collection of revision histories for about 2.7 million sentences from about 246 000 wikiHow articles. We describe human annotation studies on categorizing a subset of sentence-level edits and provide baseline models for the task of automatically distinguishing "older" from "newer" versions of a sentence.
Keywords: Corpus creation, Semantics, Other

1. Introduction

The ever increasing size of the World Wide Web has made it possible to find instructional texts, or *how-to guides*, on practically any topic or activity. wikiHow is an online platform on which a community of users collaboratively write such guides. As of March 2020, wikiHow consists of more than 246 000 articles.[1] Factors that constitute good instructional texts have been studied across various disciplines for decades, including inference requirements in cognitive science (Britton et al., 1990); document design in educational research (Misanchuk, 1992); and motivational processes in sociology (Guthrie et al., 2004). Yet, it remains open what linguistic phenomena are involved in these factors and whether they can be detected and handled automatically.

A first step towards filling this gap is to compare changes made across multiple versions of the same set of instructions, under the assumption that later versions are improvements over a first version.[2] Recent work on revisions in Wikipedia has shown that changes indeed serve a clarifying function (Faruqui et al., 2018). According to that study, however, most changes in Wikipedia provide new information (43%), with refinements only ranking second (24%).

The function and information provided by revisions is potentially different in context of how-to guides, as their content is largely independent of factual knowledge that changes over time. Therefore, we perform a study similar to that by Faruqui et al. (2018) on wikiHow articles. As our goal is to find observable patterns that reflect potential improvements, we further attempt to sub-categorize edits according to the information changed between two versions of a wikiHow article. For this step, we create a dataset of sentence-level revisions for each article in wikiHow. An example is shown in Table 1. On this dataset, we carry out two types of studies: first, we perform annotation experiments to find out more about the types of changes and proportions thereof; second, we attempt to model edits computationally by developing a system to distinguish "older" from "newer" versions of instructional sentences from wikiHow.

Text	Timestamp
1. Cut strips of paper and then write .. nouns on them (. . .)	
2. Put the pieces of paper into a hat or bag. (. . .)	
3. Have the youngest player choose the first piece of paper.	
4. Have the other players **determine** the chosen noun.	(2007-04-02T13:43:10Z)
4. Have **the** other players **guess** the chosen noun.	(2007-04-19T22:54:49Z)
4. Have **all** other players **try to** guess the chosen noun.	(2007-05-04T17:15:00Z)

Table 1: Example instruction steps from wikiHow, including different versions (and their timestamps) of the last sentence (bottom half); the example represents one of approximately 2.7 million *revision groups* in our dataset.

In summary, we make the following main contribution:[3]

- We introduce and motivate the task of distinguishing older and newer versions of instructions (§3).
- We create wikiHowToImprove, a dataset of over 2.7 million sentences and their revision histories (§4).
- We design and report on two annotation experiments that investigate the types of edits made and their proportions in a sample of revision histories (§5).
- We develop and evaluate benchmark models that distinguish different versions of a sentence (§6).

2. Related Work

In this section, we present studies conducted within two related lines of research. In Section 2.1, we discuss previous work on revisions in the English Wikipedia. Currently available wikiHow corpora are described in Section 2.2.

* equal contribution

[1] https://www.wikihow.com/wikiHow:About-wikiHow

[2] This assumption is supported, for example, by wikiHow's claim that articles are "changed 9 times per year" on average and are continually reworked "till they are the most helpful and reliable how-to guides on the web".

[3] Data and code are available here: https://github.com/irshadbhat/wikiHowToImprove

2.1. Revisions in English Wikipedia

There are a number of studies on revision histories from Wikipedia articles for various NLP tasks, such as sentence simplification and linguistic bias detection (Recasens et al., 2013). A study particularly similar to ours has been carried out by Faruqui et al. (2018) on Wikipedia articles. Faruqui et al. investigate differences between phrases inserted during a revision from the general language observed in Wikipedia texts. They approach this task through annotation experiments and linguistic analyses. The latter revealed that nouns, adjectives and adverbs occur considerably more often in edited, inserted text than non-edited text. In their computational experiments, Faruqui et al. (2018) model and analyze edits that insert information through language models based on sequence-to-sequence methods: one trained on Wikipedia texts and one trained on their own WikiEdits corpus. The task of these models is to generate a phrase which would be appropriate to insert into a sentence at a specific position. Their results show that a language model trained on article edits is more successful in proposing phrases that capture the same discourse function as human insertions than a language model trained on Wikipedia more generally. Faruqui et al. (2018) concluded that the supervision provided by article edits encodes aspects of language distinct from non-edited text.

Another study which uses the revision history of the English Wikipedia is the work of Daxenberger and Gurevych (2012). They build a corpus of 1,995 edits from 891 article revisions from English Wikipedia texts and propose a 21-category classification scheme of edit types. The categories are classified into three top layers:

- Wikipedia Policy: invalid edits as defined by internal Wikipedia Policies and respective defense mechanisms (e.g. VANDALISM).
- Surface Edits: edits not affecting the meaning of the text (e.g. SPELLING/GRAMMAR).
- Text-Base: edits affecting the meaning of the text (e.g. INFORMATION-INSERT).

Three annotators annotated the data and obtained an agreement in terms of Krippendorff's alpha (Krippendorff, 1970) of 0.67. A follow-up analysis on the frequency distribution of the type of edits showed that most edits belong to the category Text-Base (51.19%), whereas 25.64% are Surface Edits. Within the Text-Base edits, 27.34% were performed to insert, modify or delete information. Yet, it remains unknown how many of these edits were used to clarify information, since Daxenberger and Gurevych (2012) did not expose the underlying reasons to edit texts, apart from obvious ones such as spelling or grammar edits.

In a subsequent study, Daxenberger and Gurevych (2013) conducted a supervised machine learning experiment to automatically classify edits within the 21 categories. Their system scored an accuracy of 61% using language-related, textual, mark-up and meta-data related features. Examples of textual features were Levenshtein distance, token/character n-grams and the difference in number of tokens/characters.

2.2. Existing wikiHow Corpora

To the best of our knowledge, the only available corpus of wikiHow articles is from Koupaee and Wang (2018). The authors collected a large-scale summarization dataset consisting of 204,004 wikiHow articles to evaluate existing summarization systems. The structure of wikiHow articles is well suited for this task: each article is divided into paragraphs, and each paragraph starts with a summary sentence. The authors showed that the diversity of the topics and the uniqueness of n-grams (i.e., the abstraction level) in their wikiHow dataset create interesting challenges for summarization systems. For our study, the corpus of Koupaee and Wang (2018) is unsuitable since we need a collection of how-to guides that contains edited sentences as well their earlier versions.

3. Problem Statement and Motivation

The objective of this work is to categorize potential improvements made to instructional texts and to investigate in how far they can be modelled computationally. Towards this objective, we examine in how far how-to guides in wikiHow change over time. We make the simplifying assumption that changes are usually made for the better and therefore represent improvements to the original version of an article. Based on this assumption, we cast the modeling of improvements as a supervised learning problem, which requires the distinction between "older" and "newer" versions of a text. For simplicity, we focus on *edits on the sentence level*. That is, we consider all articles in wikiHow for which a revision history is available and examine each original sentence, henceforth *base version*, and how it is changed at subsequent points in time, henceforth *revised versions*.

In Section 4, we first present wikiHowToImprove, a dataset of revision histories derived from wikiHow. We describe a set of simple methods that we put together in order to automatically download and extract sentence-level revisions for articles from wikiHow. Based on a small sample of these revision histories, we attempt to categorize different types of edits in two annotation studies. These studies, presented in Section 5, provide us with potential explanations for why edits are made, thereby indicating in how far our assumption that edits represent actual improvements is reasonable. First steps to test whether such potential improvements can be modelled computationally are presented in Section 6.

4. wikiHowToImprove

wikiHow provides a collection of *how-to* articles, each describing a set of instructions to complete a procedural task. Like Wikipedia, any user can contribute to creating new entries and modifying existing ones. Further, its content is available under a Creative Commons license (BY-NC-SA). The wikiHow knowledge base contains a wide range of articles classified into 20 main categories (Arts and Entertainment, Computers and Electronics, Health, Travel etc.). Each category has a list of articles and may split into further subcategories. We exploit the wikiHow knowledge base to create wikiHowToImprove, a dataset of instructional texts along with their revision histories.

wikiHowToImprove		
Sentences, all versions	(=tokens)	6 071 010
Revision groups	(=types)	2 741 611
Word count	(=tokens)	119 664 856
Vocabulary size	(=types)	538 514
Average Sentence length		19.71

Table 2: Statistics of the wikiHowToImprove dataset.

Revision Depth	Number of Groups	Relative Frequency
1	2 283 785	83.30%
2	363 039	13.24%
3	71 307	2.60%
4	16 522	0.60%
5	4 511	0.16%
≥ 6	2 447	0.09%

Table 3: Frequency distribution over revision depths in wikiHowToImprove.

Split	Number of articles	Number of sentences	
		all versions	base only
Train	172 962	4 930 113	2 225 927
Dev	20 074	566 776	259 773
Test	19 781	574 121	255 911

Table 4: Statistics of the wikiHowToImprove data splits.

4.1. Data Collection and Corpus Creation

Unlike Wikipedia, wikiHow does not provide dumps to download, but it provides an *Export pages*[4] service which allows to export the text and full editing history of wikiHow articles in an XML. We used the python library urllib[5] to call the *Export pages* service and crawl articles with their full edit history. We obtained 246,696 unique articles at the time of crawling (20 June 2019). Each article is stored into a series of versions, separated by an XML tag `<timestamp>`. Each timestamp represents the version of an instructional text at a specific point in time.

Corpus construction. We construct a set of revised sentences in multiple steps. In the first step, we extract sentence-level edits by comparing the contents of each timestamp of an article with that of the subsequent timestamp. Within the contents of a timestamp, we remove XML tags and wiki markup, tokenize the text, and split paragraphs into sentences.[6] Before comparing sentences between two timestamps, we reduce the search space by removing sentences that have remained identical and sentences with a proportion of less than 25% English word tokens.[7] Afterwards, we follow the similarity computation by Faruqui et al. (2018) and calculate pairwise BLEU scores (Papineni et al., 2002) between each remaining sentence s_i in one timestamp and all the remaining sentences of the subsequent timestamp. We consider the sentence s_j with the highest similarity in terms of BLEU score as a revised version of the sentence s_i. If the difference between s_i and s_j is more than a case change and the similarity is greater than a threshold (0.3), we add the pair $\langle s_i, s_j \rangle$ to wikiHowToImprove. Finally, we arrange identified pairs into *revision groups*, such that each group contains the base version of a sentence and all revised versions from the subsequent timestamps in chronological order. An example revision group from the wikiHow article *"How to Play Charades"* is given in Table 1 (page 1).

Filtering cases of vandalism. Since wikiHow is a collaborative online community, anyone can contribute and not all edits are relevant to the article content. Therefore, it is necessary to identify and filter irrelevant edits. Fortunately, such edits are usually spotted and reverted back by moderators or other contributors within a few timestamps. In cases where the reversion happened within 5 timestamps, we remove the affected intermediate versions from the respective revision groups in our data. If a revision group is left with only one version, we discard this group from the data.

4.2. Data Statistics

The result of the data collection procedure described in Section 4.1 is a set of 2,741,611 revision groups. Table 2 shows the corpus statistics of the wikiHowToImprove dataset.

In the remainder of this section, we briefly describe statistics related to *revision depth*. We define the revision depth of a revision group to be the number of revised versions within that group, i.e. the number of versions of a sentence excluding the base version. For instance, in Table 1, the base version is the sentence with the first timestamp in chronological order, 2007-04-02T13:43:10Z, and the other sentences are revised versions. Consequently, the revision depth of the example shown is two.

The revision depth in our corpus varies from 1 to 33, with a mean of 1.21 and a standard deviation of 0.55. The frequency distribution over revision depths is shown in Table 3. Specifically, 83.30% (*N*=2,283,785) of all revision groups have a revision depth of 1. The shape of the frequency distribution depicts a long-tail distribution, as only a small proportion of the revision groups have a revision depth higher than 2. Furthermore, there was only one case in our corpus with a revision depth of 33.

In 384,936 cases with a depth of 1 (17.6%), we found changes to have an edit distance of 1, meaning that only one character was added, deleted or modified. We present analyses on a sample of cases with higher edit distance in the next section.

For our computational experiments, we divide the data by article into training, development and test sets using a random 80%/10%/10% split (for details, see Table 4).

[4] https://www.wikihow.com/index.php?title=Special:Export&action=submit

[5] https://docs.python.org/2/library/urllib2.html

[6] https://github.com/irshadbhat/polyglot-tokenizer

[7] according to the python library pyenchant, https://pypi.org/project/pyenchant/

Revision Type	Example
Spelling / Grammar	(1) If **possiable** pick a Leave-In-Conditionor to keep your curls tight. (1') If **possible** pick a Leave-In -Conditionor to keep your curls tight.
	(2) Try understand the movements. (2') Try **to** understand the movement.
Paraphrase	(3) **Firstly**, you create a new tradeline. (3') **First**, you create a new tradeline.
Information Deletion	(4) **Next** pick out a nice outfit . (4') Pick a nice outfit.
Information Modification / Insertion	(5) . . . it makes **them** happy to know you're getting educated. (5') . . . it makes **your parents** happy to know you're getting educated.
	(6) It 's always hard to turn down a delicious sweetened baked good. (6') It 's always hard **for adults and children** to turn down a . . . baked good.

Table 5: Overview of the different types of changes from the base version of a sentence to its revised version. Differences are highlighted in bold.

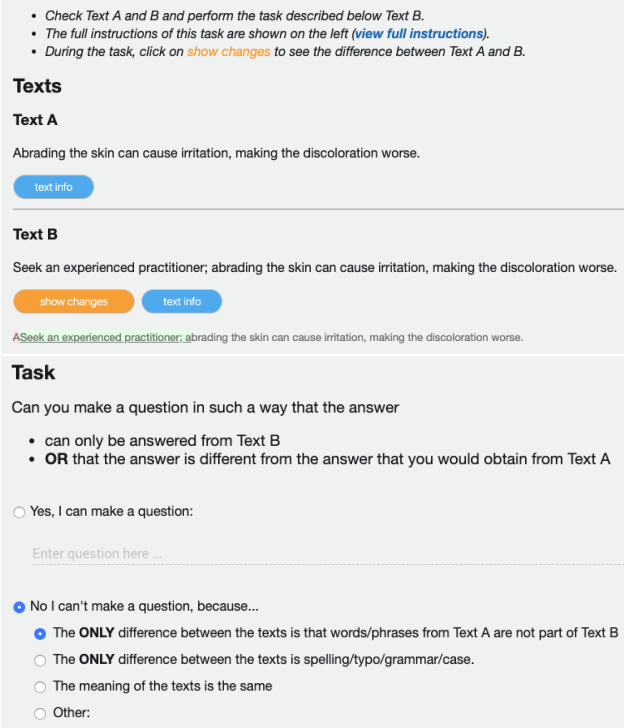

Figure 1: A screenshot of the interface, showing the base version (Text A) and the revised version of a sentence (Text B). The button 'show changes' highlights differences between both versions. The lower half of the interface shows the form that participants used to submit the question or to select a reason why they could not come up with a question.

5. Corpus Quality and Annotation Experiments

In this section, we describe a set of human annotation experiments conducted to explore the types of revisions and their proportions in wikiHowToImprove. For this purpose, we draw from the development set a sample of 100 revision groups with a revision depth of one (i.e., each instance is a pair of a base version and a revised version of a sentence) and a minimum character-based edit distance of 3. We categorize each instance in this sample according to the type of revision performed. Following this initial categorization, we perform follow-up annotations to validate and sub-categorize one of the revision types. We describe the former in Section 5.1 and the latter in Section 5.2. In Section 5.3, we summarize our findings and provide a brief discussion.

5.1. Revision Type Categorization

Our initial categorization consists of four categories, based on a manual inspection of the sample of 100 pairs of base and revised versions of a sentence. We provide examples for each category in Table 5. Since we aim to explore the proportion of edits that have likely led to improved instructions, we are mainly interested in the samples categorized as *Information Modification/Insertion*. In Daxenberger and Gurevych (2013), such edits are called text-base edits. They are of particular interest in this work because they may clarify information from the base version of a sentence.

Task. Since we are mainly interested in instances categorized as *Information Modification/Insertion*, we designed an intuitive task which should help participants to differentiate between *Information Modification/Insertion* and the other categories listed in Table 5. We initially attempted to set this up as a labeling task by providing participants labels and definitions. However, we found it difficult for annotators to pinpoint differences between versions where edits modify or provide new information, in contrast to providing only stylistic changes. Therefore, we designed the task such that annotators had to formulate questions on the presented information whenever possible. More specifically, we provided annotators with pairs of base and revised versions of a sentence, highlighted the difference(s), and asked them to formulate questions on these differences.

We required questions to be phrased such that either *the answer can be derived from both sentences, but the answers are different* or *the answer can only be derived from the revised version*, in order to ensure that they ask about information that was indeed modified or inserted. Two examples given in Table 5 illustrate the both cases: given the question "Who will you make happy?", the answer would be 'them' in (5) and 'your parents' in (5'). Given the question "For who is it difficult to turn down a delicious sweetened baked good?", the answer 'for adults and children' can be derived from (6') but not from (6).

We depict the interface that we used to ask participants to come up with a question in Figure 1. If participants could not formulate a question, then they had to indicate why this was the case by selecting one of the three reasons that we presented in the interface. Each description matched one of the definitions of *Spelling/grammar*, *Paraphrase* and *Infor-*

	Ann. 1	Ann. 2	α-score
Spelling/Fluency	9	10	0.539
Information deletion	21	17	0.741
Information Modification/Insertion	45	58	0.665
Same Meaning	25	9	0.322
Other	0	6	-
Total	100	100	

Table 6: Frequency distributions over the different revision types, as categorized by our two annotators.

mation Deletion. Furthermore, we instructed annotators to always formulate questions (i.e., apply the category *Information Modification/Insertion*) if at least one piece of information was added or changed. We included this instruction because the revised version can contain multiple parts that were modified for different reasons.

Results. We asked two students of computational linguistics to provide annotations on the 100 base–revised sentence pairs in this experiment. The results are shown in Table 6. One annotator formulated a question for 45 of the 100 cases, and the other for 58 of the cases (i.e., they assigned the category *Information Insertion/Modification*). The inter-annotator agreement of this category is $\alpha = 0.665$ (Krippendorff, 1970). In 43 cases, both annotators phrased a question.

Beyond our predefined categories, one student annotated six cases as *other*: half of them because multiple labels were applicable (*Paraphrase* and *Information Deletion*) and the other half because the base version and revised version of a sentence did not match.

5.2. Validation and Sub-categorization

In a second round of annotation, we validate instances labeled as *Information Insertion/Modification* and attempt to sub-categorize them based on the provided questions and potential answers. For this purpose, we use the set of all instances for which at least one annotator provided a question (N=60).[8] Given the small size of the sample, we asked one annotator to label 100 additional base–revised sentence pairs. This way, we collected 51 additional questions, for a total of N=111. We first provide an analysis of these questions in Section 5.2.1. We then investigate in Section 5.2.2 in how far differences in potential answers to a question reflect improvements between the base version of a sentence and its revised version.

5.2.1. Question Categorization

In a first step, we categorized the 111 collected questions into semantic classes. For this purpose, one of the authors annotated the questions using the classification scheme of Li and Roth (2002). This classification scheme categorizes questions according to the type of answer required (i.e., the information added or modified in a revised version).

The different classes and their frequencies are shown in Table 7. As indicated by the numbers, most questions are classified as DESCRIPTION (N=72). The majority of these questions were descriptions of manner. Two examples of such questions are "What should you do with the 'Brightness' bar?" and "What should be done to the pages you want to delete?". This high proportion is perhaps unsurprising given that we work with instructional texts. However, there are also a substantial number of questions asking for a certain entity. Examples of such questions are "What does marketing oversell?" and "Which dance genre is the windmill a move of?". These questions seem to imply changes in factual descriptions, rather than changes in instructions.

Main Class	Subclass
DESCRIPTION (N=72)	Manner (N=55), Reason (N=8) Description (N=9)
ENTITY (N=23)	Other (N=8) Substance (N=1) Creative (N=1), Product (N=4) Food (N=1), Event (N=6) Disease and Medicine (N=2)
NUMERIC (N=7)	Period (N=2), Other (N=2) Money(N=1), Count(N=1), Weight(N=1)
LOCATION (N=3)	Other
HUMAN (N=4)	Individual (N=2), Group (N=2)

Table 7: Frequency distribution over types of questions following the classification scheme of Li and Roth (2002).

5.2.2. Answer Collection and Analysis

In this step, we obtain a more fine-grained categorization of changes made that potentially modify or provide new information. As a starting point, we use the 111 pairs of base and revised versions labeled as *Information Insertion/Modification* from the previous round of annotation and split each pair into a *base condition* and a *revised condition*. The purpose of this setup is two-fold. First, we want to analyze the differences between the base version of a sentence and its revised version by comparing the answers provided in each condition independently. Secondly, we can use the answers within a condition to see if they are consistent or if they represent different interpretations. In each condition, we show the context of a sentence (from the respective timestamp) in addition to the sentence itself to reduce the effect of purely superficial changes.

For example, consider the sentences shown in (5) and (5'): if there is only one and the same antecedent for the pronoun 'them' and the possessive noun phrase 'your parents' in the discourse context, the answers for the question "who makes it happy to know that you are getting educated?" should be the same in both conditions. We defined as context all the sentences around the target sentence belonging to the same bullet point, enumeration point, or step in a section.

Task. For each condition, we set up a Human Intelligence Task (HIT) on Amazon Mechanical Turk. In each task, we showed participants the sentence in its base or revised version, which we highlighted, together with its respective context. We additionally showed the name of the section

[8] If both annotators provided questions, we pick one randomly.

and the title of the article. Below the text, we showed participants the question which was submitted by a participant during the categorization phase described in Section 5.1. We gathered five individual answers for each question. Thus, we collected 555 individual answers in each condition. We paid each participant $0.05 per item (question answered).
In an intermediate step, we found and discarded seven ambiguous questions, which had different answers within a condition, depending on the sentence or sentence part considered by a participant.

Results. When we analyzed the 104 remaining questions and their 520 answers in each condition independently, we observed that 475 answers followed our guidelines in the base condition. In the revised condition, the number is 481. Regarding interpretation variety, 59 out of 104 (57%) questions evoked the same individual answer in the base condition.[9] In the revised condition, there were 88 out of 104 (85%) questions that evoked the same individual answer. This indicates a potentially lower variety in interpretations in the revised condition compared to the base condition.
Still, there seem to be a considerable number of cases that involve different answers even in the revised condition. Future research will have to investigate if this is due to a need for further sentence-level improvements and/or whether the different answers can be explained by external factors (e.g., contextual paraphrases).
Across conditions, we took a closer look at those questions that received identical answers in the revised condition, but potentially different answers in the base condition. Our goal here is to identify and sub-categorize potential cases of improvement, for example, corrections and clarifications. As *potential corrections*, we consider all groups of answers that are identical in the base condition and in the revised condition, but possibly different across conditions (N=45). As *potential clarifications*, we consider all groups that contain only identical answers in the revised condition, but different answers within the base condition (N=31).
For each instance, we compare answers across conditions and sub-categorize potential differences based on changes in instruction or additional information. In case of a potential clarification, we simply compare the answer from the revised condition with the most frequent answer from the base condition. As a result of this procedure, we determine the following five main categories:

- Extension (18/45 and 18/31): the answer in the revised condition is an extension of the answer in the base condition, including words that were not part of the highlighted sentence in its base version (e.g., *press the button → long-press the button*)
- Modification (13/45 and 4/31): an instruction (or details thereof) given in the base version of a sentence is different from the revised version (e.g., *weigh . . . once per day → weigh . . . once per week*)
- Referring Entity (3/45 and 4/31): the answers refer to the same entity, but they are different in wording because different referring expressions are used in both conditions (e.g., *them → your parents*, see Table 5)
- Paraphrase (8/45 and 3/31): the answers between conditions are the same, meaning that the only difference between the two versions were syntactic differences or the usage of paraphrases. In these cases, a participant incorrectly formulated a question during the question collection procedure described in Section 5.1.
- Generalization (3/45 and 2/31): the instruction given in the revised condition is less specific than in the base condition (e.g., *smack your dog → punish your dog*)

The results show that most potential corrections (40%) and clarifications (58%) are categorized as *Extension*. In the latter cases, it is likely that participants were unable to provide identical answers in the base condition because relevant information needed to be explicitly inserted. Furthermore, 13 out of 45 (29%) potential corrections were classified as modification. In comparison, only 4 out of 31 (13%) potential clarifications were classified this way. Beyond these observations, we found no substantial differences between the distributions over the five categories for the two considered cross-condition settings.

5.3. Summary and Discussion

In our annotation studies, we considered a set of 100 sentences from wikiHow articles and examined how they changed from their base to a revised version. The results of the first annotation study indicate that differences can be grouped into four major categories: spelling/fluency improvements (∼10%), paraphrases (9–25%), information deletion (∼20%), and information modification/insertion (45–58%). In a follow-up annotation study, we sub-categorized the latter category in order to better understand the (potential) reasons behind an insertion or modification of information. We framed this study as a question–answering task, which made it possible to collect judgments for the base and revised version of a sentence independently. In an analysis across the two conditions, we found 45 cases of "potential corrections" (i.e., answers changed across conditions) and 31 cases of "potential clarifications" (i.e., answers differ in the base condition but not in the revised condition).
In a sub-categorization of potential corrections and clarifications, we found most edits (18/45 and 18/31) to be cases of added information (*Extension*). Cases of changed information (*Modification*) form the second largest sub-category (11/45 and 4/31). Although our categories and numbers are not directly comparable to those reported by Faruqui et al. (2018), they roughly correspond to the most frequent edit categories observed by them in Wikipedia.

6. Computational Experiments

In this section, we explore changes in how-to guides from a computational modelling perspective. Our annotation study has shown that revisions made between two versions of a sentence can be grouped into different categories. To investigate in how far instances of these categories can be predicted automatically, we cast the modelling of revisions

[9] For simplicity, we treat cases of "cannot answer" as different answers and cases of string overlap as identical answers.

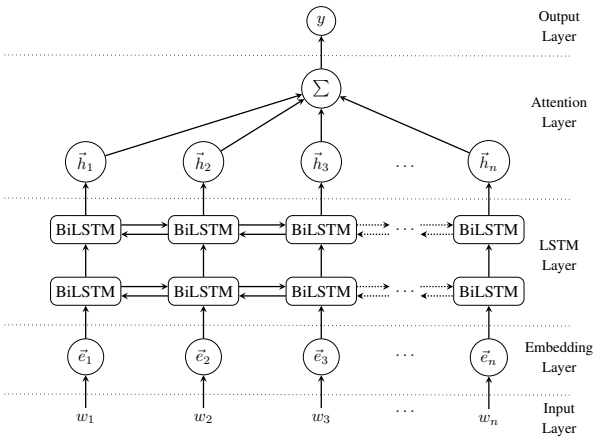

Figure 2: Bidirectional LSTM network with Attention.

as a supervised learning problem, in which the task is to distinguish between an "older" and a "newer" version of a sentence.

Task. Given a sentence pair $\langle s_i, s_j \rangle$ from a revision group, the goal of this task is to predict which is the older version and which is the newer one. For each revision group $(s_1, s_2, \ldots, s_n)$, we generate pairs of versions $\langle s_a, s_b \rangle$ as $\{\langle s_1, s_2 \rangle, \langle s_2, s_3 \rangle, \ldots, \langle s_{n-1}, s_n \rangle\}$, where s_a is an older version and s_b is a newer version.

In order to predict if a version is older or newer chronologically, we build two types of models: binary classification and pairwise ranking models. For binary classification, we convert pairs of older and newer versions into a binary dataset, labelling the first version in each pair as 0 (*old*) and the second version as 1 (*new*). One caveat of this binary task is that intermediate versions $s_2 \ldots s_{n-1}$ will appear as *old* and as *new* in different pairs. Therefore, both labels are applicable to these versions, which could make inference challenging. At evaluation time, we tackle this issue by computing continuous scores and comparing the predicted scores $\langle p_a, p_b \rangle$ for each pair of versions $\langle s_a, s_b \rangle$. If the predicted score p_b is greater than p_a, we count the prediction as correct, otherwise as incorrect. In line with this evaluation setup, we train a pairwise ranking model on pairs of versions $\langle s_a, s_b \rangle$ with the objective of learning to rank s_b higher than s_a.

We train two types of classification models: As a baseline for this task, we train a Multinomial Naive Bayes classification model that uses simple n-gram ($n = 1, 2$) features. The other type of model is based on long short-term memory (LSTM) networks, which make it possible to model sequential dependencies.

LSTM Details. We implement each LSTM-based model as a bidirectional LSTM network with an additional attention layer (Zhou et al., 2016), as illustrated in Figure 2. We use two 256-dimensional stacked BiLSTMs with a 128-dimensional attention layer on top to encode contextual information spread across the sentence. The input layer of the BiLSTM network is initialized with 300-dimensional pretrained FastText word vectors (Grave et al., 2018). The attention layer takes the BiLSTM hidden representations as input and returns their weighted sum as a embedding vector for the full sentence. For the classification model, the output layer uses a 128-dimensional feed-forward neural network with a softmax loss function. For the pairwise ranking model, the output layer on top of the BiLSTM layers uses a trained parameter vector v with a margin-based loss function:

$$\mathcal{L} = max(0, v^T \phi(s_a) - v^T \phi(s_b) + 1),$$

where $\phi(s)$ is an embedding vector from the BiLSTM network that represents sentence-level features and v^T is a transpose (i.e., row representation) of the trained parameter vector.

Model	Training	Accuracy (%)
Naive Bayes	Classification	60.80
BiLSTM	Classification	67.31
BiLSTM	Pairwise Ranking	**74.50**

Table 8: Version distinction accuracy on the test set.

Is your child **aggravating being disobedient?** Is your child **being aggravating or disobedient?**
Remember principle of 3-3-3. Remember **the** principle of 3-3-3.
Don't start **big,** start with simple and easy improvements. Don't start **big;** start with simple and easy improvements.
Never get complacent around a mother beef cow and her **call.** Never get complacent around a mother beef cow and her **calf.**
Rest the puppy **in** its back. Rest the puppy **on** its back.
Seek out of the advice **of** older relatives. Seek out of the advice **from** older relatives.
Depress the clutch fully. **Release** the clutch fully.
Keep **water** and food away from your laptop. Keep **drinks** and food away from your laptop.

Table 9: Example pairs of versions where the baseline fails, but the LSTM-based models assigns labels correctly.

Results. Table 8 shows the accuracy of each model. We see that the BiLSTM binary classification model outperforms the baseline model by 6.51% absolute accuracy. This indicates the importance of contextual information within the sentence for this task. Table 9 shows some pairs of versions where n-grams fail to capture context-dependent relationships, resulting in incorrect predictions compared to the LSTM-based models. The BiLSTM pairwise ranking model outperforms the binary BiLSTM classifier by 7.18% absolute accuracy and the baseline model by 13.70% absolute accuracy. We speculate that this could be due to the ranking mechanism's ability to implicitly model information related to transitivity between pairs of versions, which remains unobserved in the binary setting.

Figure 3 shows the average accuracy of our BiLSTM ranking model at different steps in the revision history for all groups with a revision depth between 1 and 6. Differences in results indicate that it is easier to distinguish between versions earlier in the history than later. The accuracy keeps

Revision	Text
s_0	If you **vandize** Wikipedia they will revert your edits.
s_1	If you **vandalize** Wikipedia they will revert your edits.
s_2	If you vandalize Wikipedia they will revert your edits **for preventing vandalism, generally within a few minutes or even under 10 seconds.**
s_3	If you vandalize Wikipedia they will revert your edits for preventing vandalism.
s_4	If you vandalize Wikipedia they will revert your edits **to prevent** vandalism.
s_5	If you vandalize **Wikipedia, other users** will revert your edits to prevent vandalism.

Table 10: Example revision group from the article 'How to Stop Vandalizing Wikipedia'.

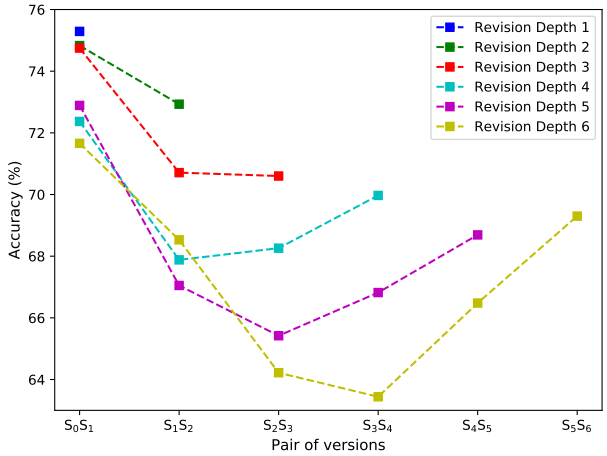

Figure 3: BiLSTM ranking accuracy by revision depth.

decreasing over time, but oftentimes performance goes up again for the distinction between the penultimate and final version. We analyze a random sample of revision groups for which this pattern holds and observe that: 1) early edits mostly fix spelling mistakes, case, or simple grammatical errors, which are easy to detect; 2) intermediate versions are most difficult to distinguish, because changes made in them are mostly due to the addition or deletion of information for stylistic reasons or due to the need for subtle semantic refinements; 3) final edits usually improve clarity by resolving ambiguities and other potential issues. Therefore, they can be slightly easier to rank than some of the intermediate versions. Table 10 shows an example revision group reflecting the aforementioned pattern. Difficulties in distinguishing versions are also correlated with how a sentence changes on the surface: as shown in Table 11, it is easier to identify newer versions when edits only add or replace text parts than when edits remove text parts. This is likely because our models cannot distinguish between text parts that provide new/relevant information from "removable" parts (e.g., information that is redundant or irrelevant).

Furthermore, inspecting the predictions of our best performing model on instances from our annotated sample, described in Section 5, revealed a high accuracy on potential clarifications (83.87%). However, the accuracy on potential corrections was slightly lower (71.11%) than the overall accuracy of the best model (74.50%). This could be because corrections contain factual changes for which integrating world knowledge into the model would be necessary.

Category	Accuracy (%)	Count of pairs (train set)
Mixed	73.48	1 201 161
Delete only	52.18	304 815
Insert only	85.46	463 180
Replace only	78.79	735 024

Table 11: BiLSTM ranking accuracy by edit category.

7. Conclusions

We introduced a corpus of sentence-level revision histories extracted from wikiHow, with the goal of categorizing and modelling potential improvements. In our first annotation study we found that revisions are most frequently made in order to describe individual steps in more detail or give additional factual information about relevant entities.

In a second annotation study, we attempted to shed light on the reasons underlying extensions and modifications by checking whether different interpretations (answers) are provided for a given text (question), depending on whether a sentence (and its context) is shown in its base or revised version. We found the average number of different interpretations to be higher for base versions, but even a revised version can still evoke different answers.

Finally, as a first step towards modeling improvements computationally, we introduced the task of distinguishing "older" from "newer" versions, which we cast as a supervised learning problem. We developed several benchmark models that employ classification and ranking methods. In an evaluation on sentence-level revisions, we found our benchmark models to achieve accuracy scores of up to 74.5%. In our analyses of results, we found that our best model is able to identify and exploit various properties related to differences in versions, including spelling mistakes and grammatical errors, but also more subtle semantic differences. On the sample from our annotation study, we found the best performing model to classify potential corrections and clarifications with high accuracy.

As next steps, we plan to set up follow-up experiments to compare results across datasets (e.g. edits in wikiHow vs. Wikipedia) and to extend our experiments and analyses to document-level settings.

Acknowledgements

The research presented in this paper was funded by the DFG Emmy Noether programme (RO 4848/2-1).

8. Bibliographical References

Britton, B. K., Van Dusen, L., Glynn, S. M., and Hemphill, D. (1990). The impact of inferences on instructional text. In *Psychology of learning and motivation*, volume 25, pages 53–70. Elsevier.

Daxenberger, J. and Gurevych, I. (2012). A corpus-based study of edit categories in featured and non-featured Wikipedia articles. In *Proceedings of COLING 2012*, pages 711–726, Mumbai, India, December. The COLING 2012 Organizing Committee.

Daxenberger, J. and Gurevych, I. (2013). Automatically classifying edit categories in Wikipedia revisions. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 578–589, Seattle, Washington, USA, October. Association for Computational Linguistics.

Faruqui, M., Pavlick, E., Tenney, I., and Das, D. (2018). WikiAtomicEdits: A Multilingual Corpus of Wikipedia Edits for Modeling Language and Discourse. In *Proc. of EMNLP*.

Grave, E., Bojanowski, P., Gupta, P., Joulin, A., and Mikolov, T. (2018). Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Guthrie, J. T., Wigfield, A., Barbosa, P., Perencevich, K. C., Taboada, A., Davis, M. H., Scafiddi, N. T., and Tonks, S. (2004). Increasing reading comprehension and engagement through concept-oriented reading instruction. *Journal of educational psychology*, 96(3):403.

Koupaee, M. and Wang, W. Y. (2018). Wikihow: A large scale text summarization dataset. *CoRR*, abs/1810.09305.

Krippendorff, K. (1970). Estimating the reliability, systematic error and random error of interval data. *Educational and Psychological Measurement*, 30(1):61–70.

Li, X. and Roth, D. (2002). Learning question classifiers. In *Proceedings of the 19th International Conference on Computational Linguistics - Volume 1*, COLING '02, pages 1–7, Stroudsburg, PA, USA. Association for Computational Linguistics.

Misanchuk, E. R. (1992). *Preparing instructional text: Document design using desktop publishing*. Educational Technology.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318.

Recasens, M., Danescu-Niculescu-Mizil, C., and Jurafsky, D. (2013). Linguistic models for analyzing and detecting biased language. In *ACL (1)*, pages 1650–1659. The Association for Computer Linguistics.

Zhou, P., Shi, W., Tian, J., Qi, Z., Li, B., Hao, H., and Xu, B. (2016). Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 207–212.