Towards Modeling Revision Requirements in wikiHow Instructions

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

Abstract

wikiHow is a resource of how-to guides that describe the steps necessary to accomplish a goal. Guides in this resource are regularly edited by a community of users, who try to improve instructions in terms of style, clarity and correctness. In this work, we test whether the need for such edits can be predicted automatically. For this task, we extend an existing resource of textual edits with a complementary set of approx. 4 million sentences that remain unedited over time and report on the outcome of two revision modeling experiments.

1 Introduction

Instructional texts have become an integral part of our daily lives, be it in the form of assembly instructions, product leaflets, troubleshooting guides, or board game manuals. A key property across all types of such texts is that they must be clear enough so that readers can actually achieve the goal described by the instructions.

Previous studies in computational linguistics have dealt with the clarity of specific types of instructions, such as route directions (Byron et al., 2009; Striegnitz et al., 2011) and software requirements (Willis et al., 2008; Yang et al., 2010). As an indicator for clarity, they relied either on successful execution in a virtual environment or on manual annotations of predefined ambiguity types. A large and more general dataset of instructional texts, wikiHowToImprove, has recently been introduced by Anthonio et al. (2020). wikiHowToImprove consists of edits for about 2.5 million sentences derived automatically from revision histories of wikiHow, a collaboratively edited platform of how-to guides. In a set of human and computational experiments, Anthonio et al. (2020) show that such edits are often made to clarify or correct a sentence and that the difference between an "older" and "newer" version of a sentence can be predicted computationally. We address two notable questions, using the work of Anthonio et al. (2020) as a starting point:

(1) Are results for the task of distinguishing two versions of a sentence (henceforth *version distinction*) specific to instructional texts, such as their guides from wikiHow, or can underlying linguistic characteristics be modelled to a similar extent in a different text genre? We reproduce version distinction results for different computational models on a variant of wikiHowToImprove and provide comparison results on an earlier dataset of revision edits (WikiAtomicEdits) derived from Wikipedia (Faruqui et al., 2018). In our experiments, we find that models for version distinction work best on instructional texts and that they are capable of detecting a variety of potential reasons for revision, including grammatical errors, semantic implausibilities and vague expressions.

(2) Given the results on instructional texts, is it possible to model whether a sentence requires revision in the first place? wikiHowToImprove only contains edited sentences. We extend the dataset with sentences from the revision history that remain identical over time. Based on this extension, we introduce the task of *predicting revision requirements* and assess its feasibility by testing whether models can distinguish sentences that get edited from ones that remain unedited. Our results show that it is possible to identify sentences that are subject to revision with a F_1-score close to 70%, indicating potential utility for downstream applications such as grammar correction (Yuan and Briscoe, 2016), ambiguity detection (Gleich et al., 2010), and machine translation refinement (Novak et al., 2016).

In summary, we make the following contributions:[1] First, we extend work on version distinction by providing experimental comparisons on wikiHow and Wikipedia. Second, we motivate a new

[1]Data and code of this work are available here: https://github.com/irshadbhat/wikiHow_MoRR

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 8407–8414,
November 16–20, 2020.

	wikiHow	Wikipedia
Sentence count	5,852,222	46,180,374
Word count	110,210,970	1,162,973,924
Vocabulary size	431,239	3,379,668
Sentence length	18.83	25.18

Table 1: Statistics of the wikiHowToImprove and WikiAtomicEdits datasets as used in Experiment 1. Counts are calculated over all versions of a text.

	subject to revision	no revision
Sentence count	4,003,412	4,258,578
Word count	75,895,857	63,546,930
Vocabulary size	406,543	308,512
Sentence length	18.95	14.92

Table 2: Statistics of wikiHow sentences that are subject to revision or no revision, as used in Experiment 2.

task of predicting revision requirements, for which we provide a new dataset as well as initial results.

2 Data and Models

Our experiments make use of WikiAtomicEdits, wikiHowToImprove and an extension of the latter that adds unrevised sentences (§2.1). For computational modeling, we re-use the publicly available baselines[2] from Anthonio et al. (2020) and further test models based on BERT (§2.2).

2.1 Data

WikiAtomicEdits. Faruqui et al. (2018) released a corpus of 43 million atomic edits across 8 languages, mined from Wikipedia edit history. The dataset consists of 26M atomic insertions and 17M atomic deletions. For our experiments, we focus on the 23M atomic edits (13.7M insertions and 9.3M deletions) from the English subcorpus, which we randomly split into training, development and test sets, setting the size of development and test set roughly equal to that of wikiHowToImprove.

wikiHowToImprove. Anthonio et al. (2020) introduced a dataset of over 2.7 million sentences and their revision histories, extracted from wikiHow. As a set of *revised sentences*, they collected *revision groups*, such that each group contains the base version of a sentence and all updated versions in chronological order. For our experiments, we use the same article-level training/development/testing splits. For a direct comparison with WikiAtomicEdits, we removed sentences longer than 50 tokens and edits that were made only because of typos (see Appendix A for more details). Statistics of both datasets are shown in Table 1.

Extensions. We extend the latter dataset by extracting around 4.25 million *unrevised sentences* from the same articles that are part of wikiHowToImprove. For each article, we collect this set by identifying sentences that have remained unchanged from the article version they were first introduced until the last version of the article. Since sentences that are introduced in the last few versions are still likely to receive revisions, we use an additional filtering criterion that measure the ratio of the number of unchanged versions of a sentence and the total number of article versions.

In preliminary experiments (see Appendix B for more details), we tested ratios between 0.0 to 0.9 on the development set to find the most suitable value. The main difference we observed in these experiments was that data imbalance and noise would make it difficult for models to distinguish between sentences requiring revision and sentences not requiring revision. For our final experiments, we use a ratio of 0.75 because we found it to reduce noise to an acceptable level and led to an almost balanced set (see Table 2). Statistics of the train/dev/test split are given in the Appendix C.

2.2 Computational Models

For both tasks, we evaluate the following methods:

Baselines. We apply as baselines for our experiments the open-source implementations of the methods from Anthonio et al. (2020): a multinomial Naive Bayes classifier with simple n-gram ($n = 1, 2$) features and a bidirectional long short-term memory (LSTM) network with an additional attention layer (Zhou et al., 2016). We use the same hyperparameters as previous work.

BERT. For additional comparisons, we train new models based on BERT (Devlin et al., 2019), a multi-layer bidirectional transformer encoder which uses bidirectional self-attention to learn deep bidirectional representations. These representations can be fine-tuned using labelled data, which led to state-of-the-art results for a range

[2] github.com/irshadbhat/wikiHowToImprove

Model	Training	Accuracy (%) wikiHow	Accuracy (%) Wikipedia
Naive Bayes	Classification	58.03	51.80
BiLSTM	Classification	64.91	62.54
BiLSTM	Pairwise Ranking	72.10	60.21
BERT	Pairwise Ranking	**73.82**	65.90

Table 3: Classification results for version distinction in wikiHowToImprove and WikiAtomicEdits.

wikiHowToImprove
Pick one band that has a large fan **basis** and ... Pick one band that has a large fan **base** and ...
It **have much** tricks and ways to interact with it. It **has many** tricks and ways to interact with it.
Plan out the details your story. Plan out the details **of** your story.
WikiAtomicEdits
She ends developing feelings for Naoto. She ends **up** developing feelings for Naoto.
She **is** also worked for the Consortium. She also worked for the Consortium.
He was Palmerston Park until 1931. He was **at** Palmerston Park until 1931.

Table 4: Example version pairs where BiLSTM and BERT models assign labels correctly.

of NLP tasks (Devlin et al., 2019; Wang et al., 2018). The pre-trained BERT models are trained on large BookCorpus (800M words) and English Wikipedia (2,500M words). For both of our experiments, we use BERT-Base, cased model (12 transformer blocks, 768 hidden size, 12 attention heads and 110M parameters) fine-tuned with an additional output layer on top of BERT's final representation. In Experiment 1, we adopt the same pairwise ranking layer as used for the BiLSTM model in previous work (Anthonio et al., 2020).[3] Experiment 2 only requires binary classification, thus the output layer simply applies a linear transformation followed by a softmax.

3 Experiment 1: Version Distinction in wikiHow and Wikipedia

The aim of the first experiment is to compare models on the task of distinguishing older and newer versions of a sentence between wikiHow and Wikipedia. We make use of the same general setup as previous work (see Anthonio et al., 2020). Our hypothesis for this experiment is that distinguishing sentence versions will be easier in wikiHow than in Wikipedia, because edits in the latter provide new information more often than refinements (Faruqui et al., 2018), whereas the content of wikiHow is largely independent of world knowledge that may change over time.

Results. Table 3 shows the accuracy of our models on wikiHowToImprove and WikiAtomicEdits. In comparison to the results reported in Anthonio et al. (2020), we observe that the baseline models are approx. 2.5% less accurate. A possible explanation for this drop is that we removed (presumably easy to predict) typo-based edits from wikiHowToImprove for direct comparison with WikiAtomcEdits. The BiLSTM pairwise ranking model outperforms the BiLSTM binary classification model by 7.18% absolute accuracy in wikiHow but is 2.33% less accurate in Wikipedia. This is the case because the ranking mechanism can implicitly model information related to transitivity when there exist more than two versions of a sentence in wikiHow. In contrast, WikiAtomicEdits only contains two versions of each sentence and pairwise ranking only brings an additional overhead due to padding (see Appendix D for more details).

The best results are achieved by BERT, which outperforms the BiLSTM ranking model by additional 1.72 percentage points on wikiHow and the BiLSTM classification model by 3.36% on Wikipedia. Finally, we see that all the models perform better on wikiHow than Wikipedia and the best performing model on wikiHow is 7.91% more accurate than on Wikipedia. This confirms our hypothesis that different versions of a sentence are harder to distinguish in the WikiAtomicEdits data.

Discussion. One reason for the higher improvement of BERT on WikiAtomicEdits could be that wikiHow and Wikipedia contain texts of different genre and only Wikipedia is used for pretraining the BERT model (see Section 2.2). In a qualitative

[3] Specifically, the output layer scores BERT's final representation $\phi(s)$ of a sentence s by calculating the dot product between the individual components and a parameter vector v, which is trained using a margin-based loss function: $max(0, v^\top \phi(s_o) - v^\top \phi(s_n) + 1)$, where s_o and s_n are the older and newer version of a sentence, respectively.

wikiHowToImprove
Follow instructions given by rail operators **always**. **Always** follow instructions given by rail operators.
If people insult you, **don't act like you care.** If people insult you, **act like you don't care.**
If you roll doubles, you **go** again. If you roll doubles, you **roll** again.
Clear the dog waste **as it happens**. Clear the dog waste **immediately**.
WikiAtomicEdits
In 1996 , he moved to play for Glenrothes. In 1996 , he moved **back** to play for Glenrothes.
Foyt married Henry in 1991 and divorced in 2013. . . . and divorced **him** in 2013.
It returns an image which is automatically updated. . . . which is automatically updated **each day**.
Dice games and slot machines are forbidden. . . . are forbidden **by state law**.

Table 5: Example version pairs where the BiLSTM model fails, but BERT assigns labels correctly.

analysis of the results, we found that the BiLSTM and BERT models are able to detect typos as well as grammatically incorrect and ill-formed sentences in both data sets (see Table 4). The BERT-based model is further able to cover more subtle syntactic corrections and semantic clarifications. As exemplified in Table 5, these cases include improvements in terms of fluency and specificity, either through changes in word order or word choice (wikiHowToImprove) or through insertions of more detailed information (WikiAtomicEdits).

4 Experiment 2: Predicting Revision Requirements in wikiHow

The aim of this second experiment is to provide benchmark models for predicting whether or not a sentence requires revision. The previous experiment has shown that it is difficult to distinguish different versions of a sentence in WikiAtomicEdits (§3). Therefore, we perform this experiment only on wikiHow. We make the simplifying assumption that *all* changes in wikiHow's revision history are made for the better and therefore represent needed revisions to the original version of an article. Thus, we treat all sentences that went through revision in wikiHowToImprove as requiring revision and all unrevised sentences from our extension (see 2.1) as requiring no revision. We evaluate to what extent a model correctly identifies sentences that require revision using precision, recall and F_1-score.

Model	Precision	Recall	F_1-score
Naive Bayes	56.44	**73.99**	64.03
BiLSTM	**73.32**	51.86	60.75
BERT	70.90	66.10	**68.42**

Table 6: Classification results for predicting revision requirements; all results are given in percentages and are shown for 'requiring revision' as the "positive" class.

Some look silly than others . . . (revised: Some look sillier than others . . .)
Buying a used car is a mine field. (revised: Buying a used car is like . . .)
You can even organize a elocution for that. (revised: . . . an elocution for that purpose.)
It is very healthy way of fast frying minimal oil. (revised: It is a very . . . frying with minimal oil.)

Table 7: Examples that require revision and that are identified correctly by BERT but not by the BiLSTM.

Results. Table 6 shows the results of our models. As shown in the table, the BiLSTM model outperforms the Naive Bayes model by 16.88 percentage points in precision, but only achieves a recall of 51.86%. This result indicates that contextual information within the sentence is needed for precisely predicting revision requirements, but it is not sufficient to achieve good coverage. The BERT model achieves the highest F_1-score, outperforming the Naive Bayes and BiLSTM models by 4.39 and 7.67 percentage points, respectively.

Discussion. In a qualitative analysis of results, we find that all models are capable (to various degrees) of identifying grammar errors and sentences that are semantically implausible. A selected list of correctly classified example sentences are given in Table 7. As shown in the table, the BERT-based model seems to capture more subtle issues on the semantic level than the BiSTM model, including adjective degrees ("silly" vs. "sillier") and metaphorical comparisons ("X is a Y" vs. "X is like a Y"). Quite likely, the BERT-based model can handle

	#Samples	LSTM	BERT
Grammar Error	1233	783	942
Lexical Vag.	164	27	93

Table 8: Comparison of BiLSTM and BERT model predictions for grammatical errors and lexical vagueness.

such cases better than the BiLSTM because BERT is pre-trained on large amounts of fluent and well-written text.

We further checked the performance of the BiLSTM and BERT-based models quantitatively on two specific types of cases from the 'subject to revision' category: grammatical errors and lexically vague modifiers. We automatically identify typical grammar errors[4] as well as cases where an adjective or adverb is replaced with a full phrase (e.g. "frequently" vs. "about once a week") by lexically and syntactically comparing the original sentence in the data to its revised version. Table 8 shows the counts of instances and correct predictions, indicating that both models have a reasonable recall regarding grammatical errors. BERT identifies more than three times as many cases of lexical vagueness as the BiLSTM model, but still only achieves a recall of 56.7% (93/164).

5 Related Work

Wikipedia Revisions. Revisions in Wikipedia have been leveraged for various NLP tasks, such as spelling error correction (Ehsan and Faili, 2013; Grundkiewicz and Junczys-Dowmunt, 2014; Zesch, 2012), preposition error correction (Cahill et al., 2013), paraphrasing (Max and Wisniewski, 2010), sentence simplification and compression (Nelken and Yamangil, 2008; Yamangil and Nelken, 2008), textual entailment recognition (Zanzotto and Pennacchiotti, 2010) and lexical simplification (Yatskar et al., 2010). Within this framework, a number of studies have analyzed the type of edits that authors made (Daxenberger and Gurevych, 2013, 2012; Faruqui et al., 2018; Pfeil et al., 2006; Bronner and Monz, 2012; Liu and Ram, 2011) and their intentions (Yang et al., 2017; Zhang and Litman, 2016). These studies built further upon Faigley and Witte (1981) and Jones (2008). Daxenberger and Gurevych (2013) and Yang et al. (2017) performed multi-class classification to automatically detect edit types and edit intentions respectively. Other text classification studies focused on a smaller set of revision intentions in Wikipedia, such as Recasens et al. (2013) who worked on bias/non-bias detection. Attention has also been given to distinguish between vandalism and non-vandalism (Adler et al., 2011; Harpalani et al., 2011; Potthast et al., 2008) and between factual and fluency edits (Fong and Biuk-Aghai, 2010)

wikiHow Revisions. Compared to Wikipedia revisions, wikiHow has received less attention in NLP. Apart from (Anthonio et al., 2020), there is no other work that leveraged the revision history of wikiHow articles. However, wikiHow has been used for summarization (Koupaee and Wang, 2018) and knowledge acquisition (Chu et al., 2017; Zhou et al., 2019). Others have also employed it to model procedure-specific relationships in sentences (Park and Motahari Nezhad, 2018) and underlying reasons for these relationships (Mishra et al., 2019).

Related Tasks. Afrin and Litman (2018), in a related task, worked with revisions in argumentative essays from ArgRewrite (Zhang et al., 2017). The authors trained a RandomForest classifier to predict, given an original sentence and a revised one, if the revised sentence is better than the original. Tan and Lee (2014) conducted a related study, which analyzed potential strength differences in original–revised sentence pairs in academic writing using a qualitative approach.

6 Conclusions

We demonstrated in an experimental comparison that it is easier to distinguish sentence versions computationally in wikiHowToImprove than in WikiAtomicEdits. We further introduced a new task of predicting whether a sentence requires revision and showed promising first results on specific types of revisions. As next steps, we plan to address further types of revisions and extend our experiments to document-level settings.

Acknowledgements

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

References

B. Thomas Adler, Luca De Alfaro, Santiago M. Mola-Velasco, Paolo Rosso, and Andrew G. West. 2011.

[4] We used the error types defined in the CoNLL-2013 shared task (Ng et al., 2013)

Wikipedia vandalism detection: Combining natural language, metadata, and reputation features. In *Proceedings of the 12th International Conference on Computational Linguistics and Intelligent Text Processing - Volume Part II*, CICLing'11, pages 277–288, Berlin, Heidelberg. Springer-Verlag.

Tazin Afrin and Diane Litman. 2018. Annotation and classification of sentence-level revision improvement. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 240–246, New Orleans, Louisiana. Association for Computational Linguistics.

Talita Anthonio, Irshad Bhat, and Michael Roth. 2020. wikihowtoimprove: A resource and analyses on edits in instructional texts. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 5721–5729, Marseille, France. European Language Resources Association.

Amit Bronner and Christof Monz. 2012. User edits classification using document revision histories. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '12, pages 356–366, Stroudsburg, PA, USA. Association for Computational Linguistics.

Donna Byron, Alexander Koller, Kristina Striegnitz, Justine Cassell, Robert Dale, Johanna Moore, and Jon Oberlander. 2009. Report on the First NLG Challenge on Generating Instructions in Virtual Environments (GIVE). In *Proceedings of the 12th European Workshop on Natural Language Generation (ENLG 2009)*, Athens, Greece. Association for Computational Linguistics.

Aoife Cahill, Nitin Madnani, Joel Tetreault, and Diane Napolitano. 2013. Robust systems for preposition error correction using Wikipedia revisions. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 507–517, Atlanta, Georgia. Association for Computational Linguistics.

Cuong Xuan Chu, Niket Tandon, and Gerhard Weikum. 2017. Distilling task knowledge from how-to communities. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, pages 805–814, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee.

Johannes Daxenberger and Iryna Gurevych. 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. The COLING 2012 Organizing Committee.

Johannes Daxenberger and Iryna Gurevych. 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. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota. Association for Computational Linguistics.

Nava Ehsan and Heshaam Faili. 2013. Grammatical and context sensitive error correction using a statistical machine translation framework. *Software: Practice and Experience*, 43.

Lester Faigley and Stephen Witte. 1981. Analyzing revision. *College Composition and Communication*, 32(4):400–414.

Manaal Faruqui, Ellie Pavlick, Ian Tenney, and Dipanjan Das. 2018. WikiAtomicEdits: A Multilingual Corpus of Wikipedia Edits for Modeling Language and Discourse. In *Proc. of EMNLP*.

Peter Kin-Fong Fong and Robert P. Biuk-Aghai. 2010. What did they do? deriving high-level edit histories in wikis. In *Proceedings of the 6th International Symposium on Wikis and Open Collaboration*, WikiSym '10, pages 2:1–2:10, New York, NY, USA. ACM.

Benedikt Gleich, Oliver Creighton, and Leonid Kof. 2010. Ambiguity detection: Towards a tool explaining ambiguity sources. In *International Working Conference on Requirements Engineering: Foundation for Software Quality*, pages 218–232. Springer.

Roman Grundkiewicz and Marcin Junczys-Dowmunt. 2014. The wiked error corpus: A corpus of corrective wikipedia edits and its application to grammatical error correction. In *PolTAL*.

Manoj Harpalani, Michael Hart, Sandesh Singh, Rob Johnson, and Yejin Choi. 2011. Language of vandalism: Improving Wikipedia vandalism detection via stylometric analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 83–88, Portland, Oregon, USA. Association for Computational Linguistics.

John Jones. 2008. Patterns of revision in online writing. a study of wikipedia's featured articles. *Written Communication*, 25:262–289.

Mahnaz Koupaee and William Yang Wang. 2018. Wikihow: A large scale text summarization dataset. *CoRR*, abs/1810.09305.

Jun Liu and Sudha Ram. 2011. Who does what: Collaboration patterns in the wikipedia and their impact on article quality. *ACM Trans. Manage. Inf. Syst.*, 2(2):11:1–11:23.

Aurélien Max and Guillaume Wisniewski. 2010. Mining naturally-occurring corrections and paraphrases from Wikipedia's revision history. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Bhavana Dalvi Mishra, Niket Tandon, Antoine Bosselut, Wen tau Yih, and Peter Clark. 2019. Everything happens for a reason: Discovering the purpose of actions in procedural text. *ArXiv*, abs/1909.04745.

Rani Nelken and Elif Yamangil. 2008. Mining wikipedia's article revision history for training computational linguistics algorithms. In *In Proceedings of the AAAI Workshop on Wikipedia and Artificial Intelligence: An Evolving Synergy (WikiAI), WikiAI08.*

Hwee Tou Ng, Siew Mei Wu, Yuanbin Wu, Christian Hadiwinoto, and Joel Tetreault. 2013. The CoNLL-2013 shared task on grammatical error correction. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–12, Sofia, Bulgaria. Association for Computational Linguistics.

Roman Novak, Michael Auli, and David Grangier. 2016. Iterative refinement for machine translation. *arXiv preprint arXiv:1610.06602*.

Hogun Park and Hamid Reza Motahari Nezhad. 2018. Learning procedures from text: Codifying how-to procedures in deep neural networks. In *Companion Proceedings of the The Web Conference 2018*, WWW '18, pages 351–358, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee.

Ulrike Pfeil, Panayiotis Zaphiris, and Chee Siang Ang. 2006. Cultural Differences in Collaborative Authoring of Wikipedia. *Journal of Computer-Mediated Communication*, 12(1):88–113.

Martin Potthast, Benno Stein, and Robert Gerling. 2008. Automatic vandalism detection in wikipedia. pages 663–668.

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

Kristina Striegnitz, Alexandre Denis, Andrew Gargett, Konstantina Garoufi, Alexander Koller, and Mariët Theune. 2011. Report on the second second challenge on generating instructions in virtual environments (GIVE-2.5). In *Proceedings of the 13th European Workshop on Natural Language Generation*, Nancy, France. Association for Computational Linguistics.

Chenhao Tan and Lillian Lee. 2014. A corpus of sentence-level revisions in academic writing: A step towards understanding statement strength in communication. 2.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, Brussels, Belgium. Association for Computational Linguistics.

Alistair Willis, Francis Chantree, and Anne De Roeck. 2008. Automatic identification of nocuous ambiguity. *Research on Language and Computation*, 6(3-4):355–374.

Elif Yamangil and Rani Nelken. 2008. Mining wikipedia revision histories for improving sentence compression. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, HLT-Short '08, pages 137–140, Stroudsburg, PA, USA. Association for Computational Linguistics.

Diyi Yang, Aaron Halfaker, Robert Kraut, and Eduard Hovy. 2017. Identifying semantic edit intentions from revisions in wikipedia. pages 2000–2010.

Hui Yang, Anne de Roeck, Alistair Willis, and Bashar Nuseibeh. 2010. A methodology for automatic identification of nocuous ambiguity. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, Beijing, China. Coling 2010 Organizing Committee.

Mark Yatskar, 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*, pages 365–368, Los Angeles, California. Association for Computational Linguistics.

Zheng Yuan and Ted Briscoe. 2016. Grammatical error correction using neural machine translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 380–386, San Diego, California. Association for Computational Linguistics.

Fabio Massimo Zanzotto and Marco Pennacchiotti. 2010. Expanding textual entailment corpora fromWikipedia using co-training. In *Proceedings of the 2nd Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources*, pages 28–36, Beijing, China. Coling 2010 Organizing Committee.

Torsten Zesch. 2012. Measuring contextual fitness using error contexts extracted from the Wikipedia revision history. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 529–538, Avignon, France. Association for Computational Linguistics.

Fan Zhang, Homa B. Hashemi, Rebecca Hwa, and Diane Litman. 2017. A corpus of annotated revisions for studying argumentative writing. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1568–1578. Association for Computational Linguistics.

Fan Zhang and Diane Litman. 2016. Using context to predict the purpose of argumentative writing revisions. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1424–1430, San Diego, California. Association for Computational Linguistics.

Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 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.

Yilun Zhou, Julie Shah, and Steven Schockaert. 2019. Learning household task knowledge from WikiHow descriptions. In *Proceedings of the 5th Workshop on Semantic Deep Learning (SemDeep-5)*, pages 50–56, Macau, China. Association for Computational Linguistics.

A Filtering Typos

For each revision pair (base, revised), we find the sentence edits and edit types using Levenshtein distance algorithm. If all the edits are of substitution type and every substitution fixes a typo, we remove the base sentence.

B Selection of Filtering Ratio.

In order to select an appropriate filtering ratio, we ran Experiment 2 (*Predicting Revision Requirements*) with 10 different ratios from 0.0 to 0.9. Based on the results on our validation set (see Figure 1) and the data imbalance at each ratio, we selected a ratio of 0.75, which lead to an almost balanced set. A ratio of 0.75 means that sentences have to remain "identical" for the last 3 out of 4 article-level revisions in order to be considered as "not requiring revision".

C Training/Development/Testing Split for Experiment 2

D Padding in Pairwise Ranking Models

We train all our BiLSTM models with a batch size of 512. For the BiLSTM classification model, we simply batch the sentences based on sentence length, so no padding is required. But for the pairwise ranking models, we have to batch version pairs (base, revised) together. We can only batch these pairs if the number of tokens in both versions of a sentence are equal. So we first append pads to the shorter sentence in the version pair to make its length equal to the longer sentence and then batch these pairs based on length.

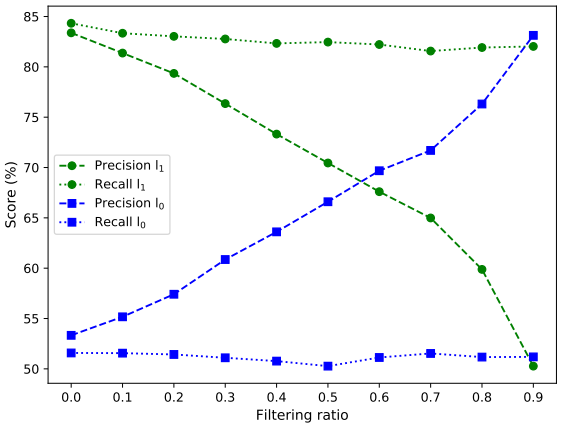

Figure 1: Scores for predicting revision requirements at different filtering ratios. l_0='requiring revision', l_1='requiring no revision'.

Split	subject to revision	no revision
Train	3 249 521	3 467 462
Dev	378 996	393 770
Test	374 895	397 346

Table 9: Statistics of the training, development and testing splits, as used in Experiment 2.