A Mention-Based System for Revision Requirements Detection

Ahmed Ruby[1], Christian Hardmeier[1,2] and Sara Stymne[1]
[1]Uppsala University, Department of Linguistics and Philology
[2]IT University of Copenhagen, Department of Computer Science
firstName.lastName@lingfil.uu.se

Abstract

Exploring aspects of sentential meaning that are implicit or underspecified in context is important for sentence understanding. In this paper, we propose a novel architecture based on mentions for revision requirements detection. The goal is to improve understandability, addressing some types of revisions, especially for the Replaced Pronoun type. We show that our mention-based system can predict replaced pronouns well on the mention-level. However, our combined sentence-level system does not improve on the sentence-level BERT baseline. We also present additional contrastive systems, and show results for each type of edit.

1 Introduction

The Revision Requirements task aims to recognize whether or not a sentence requires revision. Revision Requirements prediction not only acts as a standalone tool for grammar correction but also has potential applications in natural language processing (NLP) such as ambiguity detection, machine translation refinement, sentence understanding, knowledge base construction, etc.

The shared task on implicit and underspecified language (Roth and Anthonio, 2021)[1] aims to provide a binary classification for revision requirements to make a prediction of whether sentences in instructional texts require revision to improve understandability. Since instructional texts must be clear enough so that readers and machines can actually achieve the goal described by the instructions, this task focuses on modeling implicit elements that make the sentence more precise and clear. The dataset used in this shared task consists of instances from wikiHowToImprove, a collection of instructional texts, which has recently been introduced by Anthonio et al. (2020). It contains six types of edits:

- Replacements of pronouns with more precise noun phrases (REPLACED_PRONOUN)
- Replacements of 'do' as a full verb with more precise verbs (REPLACED_DO)
- Insertions of optional verbal phrase complements (ADDED_ARG)
- Insertions of adverbial and adjectival modifiers (ADDED_MOD)
- Insertions of quantifiers (ADDED_QUANT)
- Insertions of modal verbs (ADDED_MOD)

The shared task submission requires only a binary distinction between sentences that require revision and sentences that do not.

A good instructional text consists of specific instructions to accomplish the goal described and tends to avoid vague, generic and generalizing sentences. Whilst checking the edit types in the revised version, especially "Replacements of pronouns with more precise noun phrases", we observed that replacements occur primarily with generic pronouns that do not refer to a specific individual or set of individuals, but to a type or class of individuals. Table 1 shows examples with generic pronouns that require revision.

For this reason, we believe and show that identifying generic pronouns and noun phrases helps to predict whether a sentence requires revision for the REVISED_PRONOUN class. For instance, if the pronoun has a co-reference in the sentence, it should not be replaced with a noun phrase. As a result, our proposed classification model for the task of Revision Requirements Detection is based on extracting mention embeddings for each sentence

[1] https://unimplicit.github.io

Proceedings of the 1st Workshop on Understanding Implicit and Underspecified Language (UnImplicit 2021), pages 58–63
Bangkok, Thailand (online), August 5, 2021.

Generic pronouns
They make a sound that dogs can hear, but humans can't.
Double check that **it** will be level using a level.
Your parents may not like any of **them**.
Burn **it** to a CD.
Let **us** have bad days.
You cannot be offside directly from a corner-kick.

Table 1: Examples with generic pronouns that require revision.

using a neural coreference resolution system[2] and feed them into a classification layer (multi-layer perceptron) to predict for each individual mention whether or not it requires revision.

Our approach uses the Neuralcoref resolution system to get mention embeddings for the target sentence. In addition we also extract embeddings for each mention based on BERT (Devlin et al., 2019) for each mention. In this approach, we predicted revisions at the mention level. Labels for the mentions were created based on a comparison between the original sentence, and the revised sentence, where we checked if any word had been changed, added or removed from each mention. For sentences for which we could not extract any mentions, we used a basic sentence-level Bert-based system, since the BERT model achieved the highest F1-score in previous work (Bhat et al., 2020).

In summary, we show that our mention-based system works well for replaced pronouns, but as expected, it is not successful for the other classes, which it does not target. Our final system is overall slightly worse than our sentence-based system based on BERT. At the mention-level our system performs well for replaced pronouns.

2 Related Work

There has been a lot of work on revisions to improve understandability, Tan and Lee (2014) conducted research on revisions in academic writing, using a qualitative approach to distinguish between strong and weak sentences, by analyzing the differences in the original and revised sentences.

Afrin and Litman (2018) introduced a classification model based on Random Forest (RF) for revisions in argumentative essays from ArgRewrite (Zhang et al., 2017) to examine whether we can predict improvement for non-expert and predict if the revised sentence is better than the original.

Anthonio et al. (2020) worked with edits in instructional texts and applied a supervised learning approach to distinguish older and newer versions of a sentence between wikiHow and Wikipedia.

Recent work by Bhat et al. (2020) presents an automatic classification of revision requirements in wikiHow, used the BERT model to achieve the highest F1-score, reporting 68.42% predicting revision requirements, outperforming the Naive Bayes and BiLSTM models by 4.39 and 7.67 percentage points, respectively. We consider the BERT Model as a strong baseline for our experiments from Bhat et al. (2020).

Dataset	Req_Revision	N. of sentences
Training set	19599	39187
Development set	1632	3264
Test set		3458

Table 2: Statistics of the dataset.

3 Dataset

We used the dataset provided by the organizers of the shared task on revision requirements prediction. This dataset contains instances that were extracted from the revision histories of www.wikiHow.com articles. These how-to articles cover many fields such as Arts and Entertainment, Computers and Electronics, Health, along with their revision history. The revisions and classes were extracted automatically from the training data. The development and test data was verified by human annotators (see Roth and Anthonio, 2021, for details).

There are two subsets:

- Sentences extracted from the revision history, which later received edits which made the sentence more precise. These are labelled REQ_REVISION.
- Sentences that remained unchanged over multiple revisions of the article. These are labelled KEEP_UNREVIS.

The dataset includes training, development and test sets. However, the type of edit in case of a revision and the revised version of the target sentence, are available only for the training set. We therefore used k-fold cross-validation to randomly partition the training set into 5 equal-sized subsamples for training and development, for which we needed access to the revised sentences. Table 2 shows how the dataset is balanced.[3]

[2]https://github.com/huggingface/neuralcoref

[3]The test set is not released to participants, so we cannot report all test set statistics.

Dataset	Req_Revision	N. of mentions
Training set	2901	16976
Development set	749	4339
Test set		2368

Table 3: Statistics of the mentions in the dataset.

We used SpaCy's (Honnibal et al., 2020) tokenizer to tokenize the target sentences and the context since the current dataset does not include the tokenized version of the context.

4 Mention Extraction

Based on our observation that generic noun phrases often lead to revision, we hypothesize that extracting mentions based on a coreference resolution system might help in identifying such instances. We believe that this architecture might be especially useful for replacements of pronouns with more specific noun phrases and the insertion of logical quantifiers. We use Huggingface's NeuralCoref system, which is based on spaCy library, to extract mentions from our dataset.[4] Table 3 shows statistics of the mentions in the dataset.

In order to create labels for mentions, we extracted the class of each token for the input target sentence by comparing the target with the revised sentences. We use the Python *difflib* library to align the original and revised sentence. We can then assign a positive label if any word in a mention was removed, changed or inserted and a negative label otherwise.

5 Mention Embeddings

Since we need to capture the coreference information within the span of mentions in the embeddings, we produced two versions of the mention embeddings, one with dimension 650 using Neuralcoref resolution system and a second of dimension 768 using BERT-as-service.

5.1 Mention NeuralCoref Embeddings

We use Huggingface's NeuralCoref system as well to get embeddings for mentions, which is based on SpaCy's model en_core_web_lg. All embeddings are extracted for all mentions found in the target sentence.

[4]https://github.com/huggingface/neuralcoref.

5.2 Mention BERT Embeddings

We use bert-as-service,[5] uncased model, to generate the BERT embeddings, with our own per-mention reduction, which takes the vectors for each word and does the mean reduction for these vectors which were extracted corresponding to the span of the mention.

5.3 Concatenating the BERT and NeuralCoref embeddings

We also try using the combination of both embeddings. Therefore, we concatenated the BERT embeddings and neuralcoref vectors. the dimensions of the concatenated output vector are 1418.

6 Experimental and Model Design

In this section, we present initial exploratory experiments and the process behind building a model that addresses the two obstacles to combine the predictions of the mention-level system into the sentence-level BERT backup system.

6.1 Mention-Level System

For the mention-level, we use a feed-forward neural network with the different types of mention embedding as input to classify whether the mention requires revision or not.

We train a Multi Layer Perceptron (MLP) using mention embeddings as input, with using a single hidden layer consisting of 100 hidden units and a rectified linear (ReLU) activation function, and the final linear layer with a sigmoid function to make predictions. Since the mention dataset is not balanced, the classifier sees many more negative than positive examples. We try to counteract this by giving higher weight to the positive examples using class weights. For experiments on the training data, where we use cross-validation, the weights for the negative and positive classes, were set to 0.854 and 0.146 respectively while the weights for the full training data for the negative and positive classes, were set to 0.840 and 0.160 respectively.

We run 3 experiments with different inputs, mention NeuralCoref embeddings (M), mention BERT Embeddings (MB) and mention BERT and neuralcoref embeddings concatenated (M+MB).

All models are trained for 100 epochs and with a learning rate of 0.01, and training examples are presented in random order. For experiments on the

[5]https://github.com/hanxiao/bert-as-service

training data, where we use cross-validation, we report the average scores across the five folds.

6.2 Sentence-Level System

For the sentence-level system, we use BERT-Base (Devlin et al., 2019), uncased 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. We use the Huggingface Transformers library with TensorFlow and load a pre-trained BERT from the Transformers library. We train this model for 2 epochs with a learning rate of $3 \cdot 10^{-5}$ and batch size 32.

The mention-level system does not have extracted mentions for all sentences, and therefore does not provide predictions for all sentences. In our combined system we use the predictions from the mention-level system as our primary predictions; if there is a positive prediction for any mention in a sentence, that sentence is labelled as positive. Sentences where all mentions are labelled negative receive a negative label. For sentences without extracted mentions, we use the predictions by the sentence-level BERT-based system.

As a further point of comparison we also provide an oracle combination of the two systems. In the oracle we only use the predictions from the mention-based systems for those sentences where there is at least one mention which requires an edit, i.e. which has a positive gold label. The purpose of this oracle is to give an idea of how well our mention-based system performs on mentions where we know an edit is required. For all other sentences, the oracle uses the prediction from the sentence-level BERT-based system.

7 Results and Analysis

This section presents an overview of our experiments and findings. We compare our results with the BERT model baseline that set the previous state-of-the-art performance. We also present results on specific types of revisions since our approach was targeted mainly at the "replaced pronoun" class. We perform the majority of our analysis on training set, presented in Tables 4–7, which is the only data set which contains class labels and revised sentences. Precision, recall, and F_1-score is shown for requiring revision as the positive class.

The most successful model on mention-level is the system with only mention Bert embeddings (MB), as shown in Table 4. The system using mention Neural Coref embeddings is not successful and always predicts a single class; in all folds but one it predicts the negative class only. The difference between using only BERT embeddings and combining the two embedding types is small.

Model	Precision	Recall	F_1-score	Acc
M	0.0292	0.2000	0.0510	0.7123
MB	0.2646	0.6783	0.3799	0.6772
M+MB	0.2575	0.7273	0.3797	0.6523

Table 4: Results of our models on mention-level.

Types of revision	MB
ADDED_ARG	0.4208
ADDED_MOD	0.0578
ADDED_MODAL	0.1500
ADDED_QUANT	0.1768
REPLACED_DO	0.6492
REPLACED_PRONOUN	0.9064

Table 5: Recall for each type of edits on the mention-level

Table 5 shows the results for each type of edit, for the best mention-level system, with BERT embeddings. Here we can only show recall, since our system does not predict the individual classes. The results confirm that our system is useful for detecting the pronoun replacement class as revision requirements, but that it gives poor results for the other classes, especially for added modifiers, modals, and quantifiers.

Table 6 shows the results of our models on the sentence-level. Overall it is clear that the sentence-level BERT-based system is better than the mention-based combinations, shown in the middle, especially with respect to recall. The M+BERT system has the lowest recall. The bottom row shows the oracle scores for the MB+BERT system, which gives slightly better results than the BERT baseline on all metrics, which indicates that the decisions made by the mention-based system are good with respect to sentences where an extracted mention requires revision. The oracle is considerably better than the standard combination, especially for recall, since the mention-based system does not really have a chance to predict anything useful for sentences where the edit does not occur in one of our extracted mentions.

Table 7 shows recall for each type of edit on the sentence-level. The sentence-level BERT-based system still achieves the highest scores for all classes compared to the standard combination. The oracle combination shows an improvement for the

Model	Precision	Recall	F_1-score	Acc
BERT	0.6628	0.5997	0.6275	0.6460
M+BERT	0.6459	0.3816	0.4742	0.5847
MB+BERT	0.6470	0.4906	0.5567	0.6107
M+MB+BERT	0.6455	0.4989	0.5617	0.6118
MB+BERT oracle	0.6654	0.6064	0.6324	0.6493

Table 6: Results for predicting revision requirements at the sentence-level. The top row is the BERT sentence-level baseline, the middle rows shows the combined system, and the bottom row the oracle combination for MB embeddings.

Types of revision	BERT	MB+BERT	MB+BERT oracle
ADDED_ARG	0.7506	0.6369	0.7495
ADDED_MOD	0.4731	0.4121	0.4719
ADDED_MODAL	0.6573	0.5505	0.6563
ADDED_QUANT	0.4236	0.3588	0.4244
REPLACED_DO	0.7848	0.6737	0.7828
REPLACED_PRONOUN	0.8229	0.5856	0.8586

Table 7: Recall for each type of edit on the Sentence-level

Model	Precision	Recall	F_1-score	Acc
BERT	0.7044	0.6146	0.6564	0.6783
M	0.6590	0.5472	0.5979	0.6320
MB	0.6891	0.4522	0.5461	0.6241
M+MB	0.6831	0.4914	0.5716	0.6317

Table 8: Sentence-level results on the development set.

replaced pronoun class compared to the BERT baseline. For the other classes the difference to the baseline is small for the oracle, with only slightly lower results, which indicates that the mention-based system hardly ever predicts an edit for the other classes, and the few times it does so, it is mainly erroneous.

Table 8 shows the results on the provided development sets. These results are generated by using only the standard combination since we do not have gold labels for the mentions, since no revised sentences were provided for the development set. The BERT model achieves the highest F_1-score, outperforming the M, MB and M+MB by 4.63, 5.42 and 4.66 percentage points, while the M outperforms the MB and M+MB models by 9.50 and 5.58 percentage points in recall since the M system only predicts a single class.

We submitted the combination system based on MB embeddings to the shared task.[6] For our submitted predictions on the test, which was evaluated by the organizers in terms of accuracy, measured as the ratio of correct predictions over all data instances (Roth and Anthonio, 2021), we achieved 66.3% accuracy for the mention-based system, which is higher than the logistic regression baseline provided by the organizers. Our sentence-level BERT-based system achieved 68.6% accuracy on the test set.

[6]Our original submission had a bug, leading to low scores. We thus report results for our updated submission, without this bug, which is also reported in Roth and Anthonio (2021).

8 Conclusions and Future Work

In this paper, we show that identifying generic mentions can improve the performance of the replaced pronoun type. We introduced a mention-based system for predicting whether a sentence requires revision. Investigating methods for combining a general classifier such as BERT, with systems that target specific edits, such as our mention-based system, would be an interesting avenue for future work. As a next step, we plan to apply this idea to other languages and address other types of revisions.

Acknowledgements

Christian Hardmeier was supported by the Swedish Research Council under grant 2017-930.

References

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.

Irshad Bhat, Talita Anthonio, and Michael Roth. 2020. Towards modeling revision requirements in wikiHow instructions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8407–8414, Online. 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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Michael Roth and Talita Anthonio. 2021. Unimplicit shared task report: Detecting clarification requirements in instructional text. In *Proceedings of the First Workshop on Understanding Implicit and Underspecified Language*.

Chenhao Tan and Lillian Lee. 2014. A corpus of sentence-level revisions in academic writing: A step towards understanding statement strength in communication. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 403–408, Baltimore, Maryland. 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, Vancouver, Canada. Association for Computational Linguistics.