Revision Classification for Current Events in Dutch Wikipedia using a Long Short-term Memory Network

Nienke Eijsvogel and Marijn Schraagen

Utrecht University, The Netherlands
N.L.M.Eijsvogel@students.uu.nl, M.P.Schraagen@uu.nl

Abstract. Wikipedia contains articles on many important news events, with page revisions providing near real-time coverage of the developments in the event. The set of revisions for a particular page is therefore useful to establish a timeline of the event itself and the availability of information about the event at a given moment. However, many revisions are not particularly relevant for such goals, for example spelling corrections or wikification edits. The current research aims to classify revisions automatically given a set of revision categories, in order to identify which revisions are relevant for the description of an event. In a case study a set of revisions for a recent news event is manually annotated, and the annotations are used to train a Long Short Term Memory classifier for 11 revision categories. The classifier has a validation accuracy of around 0.69 which outperforms recent research on this task, although some overfitting is present in the case study data. The paper provides an error analysis and a discussion on the results and future steps towards the goal of timeline extraction.

Keywords: Revision classification · LSTM · Corpus annotation

1 Introduction

Wikipedia is a well-known resource for research in data mining and machine learning in general, and Natural Language Processing (NLP) in particular. The resource is used because of the size, quality and structure of the data. The Dutch language version of Wikipedia contains around 1 million articles as of July 2019 (article stubs excluded). Many pages are encyclopedic in nature, however coverage extends to current and unfolding events as well. Wikipedia articles are edited by the user community and the full time-stamped edit history is preserved. For Dutch Wikipedia the community consists of around 4,000 active users, and the pages contain a total number of around 54 million contributed edits. This allows research into the development process of the content, besides research into processing knowledge and information on the topic of the article. The development

of articles on events that are in progress at the time of writing are of particular interest, because revisions to these articles reflect the development of the event itself. Such revisions can be used to automatically extract and present a timeline of the event, as well as the availability of information at a given moment.

Wikipedia pages are edited for a variety of purposes. Some edits are syntactic or administrative, such as correcting spelling and grammar or making sure the page conforms to formatting guidelines. Other edits are more semantic in nature, such as adding facts or correcting factual mistakes. Syntactic edits are generally irrelevant for the purposes of extracting and presenting event developments, therefore such edits should be discarded. Given that categorizing the purpose of a revision is a non-trivial task, the current research proposes an approach for automatic revision classification using machine learning.

To enable a supervised learning approach a dataset consisting of annotated revisions is required. For Dutch Wikipedia no such datasets are available, therefore the current research includes an annotation effort in which a set of revision categories is established (based on existing literature) and a small corpus of revisions is annotated using this set of categories.

The paper is structured as follows: in Section 2 an overview of related work is presented, Section 3 provides details on the annotation and classification approach, the results of the experiments are presented and analyzed in Section 4 and Section 5 discusses the results and provides an outline for future work.

2 Related work

A variety of NLP tasks benefit from the near-parallel data stored in the revision history, such as paraphrasing [6], simplification [7] and error correction [3]. An approach for edit classification specifically aimed at detecting vandalism (malicious edits) is presented in [1], using both linguistic features and metadata characteristics.

The topic of general edit classification is investigated by Daxenberger and Gurevych [4]. A classification scheme containing 21 categories for edits was designed and used to construct the Wikipedia Quality Assessment Corpus (WPQAC) containing 21,578 revisions. The corpus was designed to reduce the effects of article popularity and visibility by including pairs of articles with similar properties (e.g. size, rate of daily revisions) from different classes according to the internal Wikipedia quality review system. Using this corpus, [5] constructed various machine learning classifiers for the revision categorization task. The best performing classifier was an ensemble method using a set of single-label classifiers using the C4.5 decision tree algorithm. A large number of features was used, among which textually oriented features that measure differences in the number of certain characters such as digits or whitespace, and a category called "language features" that count the number of out-of-vocabulary words, vulgar words, and Part-of-Speech tags. Daxenberger and Gurevych concluded that textual features had the highest impact on classifier performance, whereas language features play only a minor role.

A different categorization scheme was proposed by [8]. This scheme is aimed at classifying editor intentions and contains 13 categories. A corpus was constructed containing 7,177 revisions, which was used to train a Binary Relevance ensemble classifier. The experiments show that this classifier outperforms heuristic baselines as well as the ensemble classifier from [5]. The current research uses a categorization scheme adapted from [8]. The experiments differ from previous research in the use of Dutch Wikipedia instead of the English language version. Furthermore, a network is trained on vocabulary features only, using a Long Short Term Memory classifier. Moreover, the current research adds to related work by providing a new set of manually annotated page revision data.

3 Method

The research methodology consists of three stages. First, a categorization scheme is established and a case study is selected for which the set of revisions is annotated. Second, a baseline classifier is designed and implemented, to gain insight into the nature of the problem. In the final stage a Long Short Term Memory network (LSTM) is trained on the annotated data.

Table 1. Revision categorization scheme

category	*definition*	*original*	*revision*	*n*
clarification	specify or explain, no new information	24 October square	24 October square in the Kanaleneiland district	68
rephrase	rephrase without altering meaning	on March 28	ten days later	302
grammar	spelling, grammar	of on terrorist attack	of a terrorist attack	74
elaboration	add new content		a police dog tracked the suspects	37
fact update	new facts available	5 people wounded	9 people wounded	127
simplifying	delete irrelevant detail	local debate cancelled		46
vandalism	malicious edit	Central European Time	Central African Time	6
recovery	revert vandalism	Central African Time	Central European Time	6
verification	delete unverified fact	offender: Gökmen T.	offender not confirmed	119
wikification	formatting, Wikipedia links	minister Grapperhaus	minister [[Ferdinand Grapperhaus]]	66
reference	external references		Suspect confesses, `https://nos.nl`	7

3.1 Data annotation

The case study examines the page on the shooting incident in the Dutch city of Utrecht on March 18th 2019[1]. The edit history between the creation of the page

[1] https://nl.wikipedia.org/wiki/Schietincident_in_Utrecht_op_18_maart_2019

and May 16th, 2019 is taken into account, consisting of 858 revisions in total[2]. A categorization scheme using 11 classes is established, modelled after [8]. The list of labels with example revisions is provided in Table 1, showing the number of annotations n in the case study data. Compared to [8] the current scheme shows a number of differences. The original scheme contains three additional categories, which are *Process*, *Disambiguation* and *Point of View*. The first two categories deal with Wikipedia-specific technical process edits, and have been classified as *Wikification* in the current scheme. The *Point of View* category was not found in the current data and has been omitted. Finally, the original *Verification* category has been split in two categories *Reference* and *Verification*, to reflect the semantic difference between these two types of revisions. Annotations are performed by a single annotator. In case multiple categories could apply to a revision the category is selected that, in the opinion of the annotator, captures the most prominent intention of the editor.

3.2 Baseline

As a baseline a C4.5 decision tree is trained using syntactic features. The features used in the decision tree are listed in Table 2. The selection of features has been performed based on initial exploration of the data. This baseline is intended as a rule-based approach where the C4.5 algorithm selects the most important rules and the threshold values for numerical features. The comparison of a rule-oriented baseline with a machine learning approach can provide insights into the limitations of character-based rules and the added value of a neural network classifier in general, and vocabulary-based features in particular. Furthermore, the baseline can aid in positioning the performance of the current research within the results of related work.

Table 2. Syntactic features used in the baseline system

feature	*value*	*description*
insertion	binary	1: insertion, 0: deletion
text length	integer	number of characters
reference	binary	revision contains string *ref*
int	binary	revision contains an integer number
wiki	binary	revision contains characters [[ ]]
ratio	float	ratio of insertion and deletion
blocks	integer	amount of edits in different positions on the page

3.3 LSTM training

The main classifier trained on the data is a fully connected Long Short-term Memory network (LSTM). The input is represented by modelling the input text

[2] The annotated data is available on https://git.science.uu.nl/snippets/44.

of a revision with 300-dimensional word embeddings. For a small dataset as used in the case study it is not feasible to train embeddings on the data itself, therefore the pretrained Dutch fastText [2] set is used. This set is trained on Dutch Wikipedia, and is therefore expected to be suitable for the current data[3]. Given that most revisions are short (average 11.2 words, median 3 words, maximum 207 words), all revisions are cut off to 100 words to reduce the number of zero-valued entries in the input vectors. Following the embedding layer an LSTM layer and up to three dense layers are implemented. Given the multi-class nature of the problem the Categorical Cross-entropy loss function is used for the network.

4 Results

The baseline classification tree using syntactic features results in an accuracy of 0.53 on the test set. The length of the revision is the most important feature, which is selected several times by the C4.5 algorithm. Other important features are the ratio between inserted and deleted text, and the number of blocks on different positions on the page edited during a particular revision. The remaining features are less prominent in the tree, which is expected given that these features correspond to a specific category.

Table 3. LSTM Classification performance, averaged over 10 runs

dense layers	*batch size*	*training accuracy*	*test accuracy*
1	60	0.90	0.65
1	70	0.85	0.66
1	80	0.87	0.66
1	90	0.88	0.65
2	60	0.94	0.67
2	70	0.91	0.66
2	80	0.93	0.67
2	90	0.93	0.66
3	60	0.94	0.68
3	70	0.93	0.68
3	80	0.94	0.69
3	90	0.94	0.68

For the LSTM network experiments are performed with different batch sizes (60–90) and a different number of dense layers (1–3). Table 3 lists accuracy for different parameter settings. The table shows that batch size has a very small influence of the result (0.0pp–0.1pp.) while the influence of the number of dense layers is somehwat larger at a maximum of 0.4pp. The best performance is reached using a three dense layers on top of the LSTM layer with

[3] The out of vocabulary rate of the dataset relative to fastText is 13.8%, of which many OOV items are numbers or parts of URLs.

batch size=80 (Figure 1). The figure shows that the training accuracy and the validation accuracy deviate after approximately 20 epochs, indicating overfitting on the training data. However, validation accuracy remains stable at ∼0.69. The related work in [8] shows an accuracy of 0.54 for the best performing classifier, using 13 categories on a dataset of 5,000 revisions randomly sampled from English Wikipedia. Because of the different dataset and the modifications of the categorization scheme, the results are not directly comparable. However, the result shows that the current method results in competitive performance compared to related work.

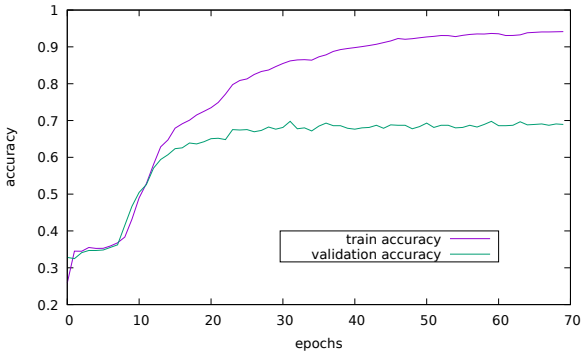

Fig. 1. Classification performance (3 dense layers, batch size 80), 10 runs average

4.1 Error analysis

The errors made by the classifier (on both training and test data) are shown in Figure 2. Notable misclassifications occur for the *wikification* class and the *fact update* class. Actual wikification revisions often consist of inserting only brackets or inserting words within brackets. Examples of this class are regularly misclassified as *rephrase*, *grammar* or *fact update*. This can be explained by the characteristics of these three classes: fact update and grammar both use a small number of characters, and rephrase is similar to wikification especially when multiple links are added. The *fact update* class is predicted several times for a number of other classes, which can be explained by the syntactic resemblance to *grammar* and *wikification*. If a fact update is done in sentence for the revision resembles *rephrase* examples. Furthermore, *rephrase* is confused several times with *clarification* and *simplifying*. This is caused in part by the input representation, which is discussed in the next section. Finally, *elaboration* can be classified incorrectly as *reference* since adding a substantial amount of text often comes with new information that has a reference link. In such cases the revision is labelled as *reference*, as this is considered the most important editing intention.

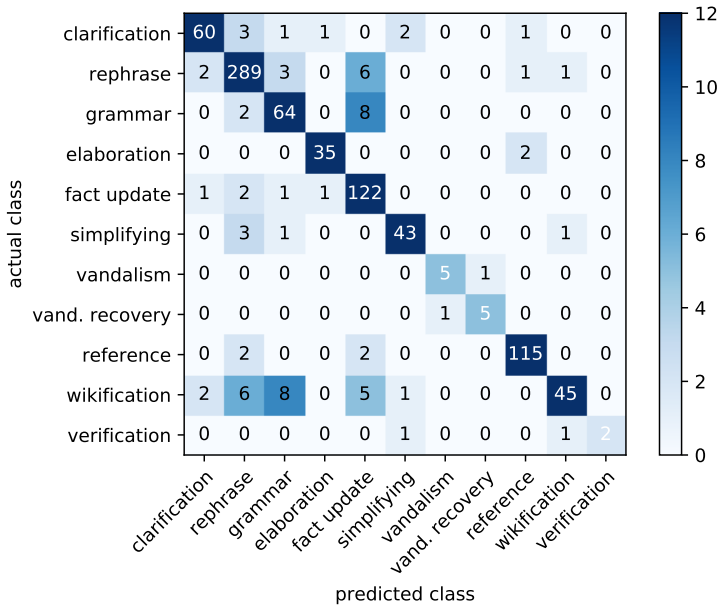

Fig. 2. Classification confusion matrix

5 Discussion and Future Work

The LSTM network produces a number of misclassifications that are caused by the input representation. In case a page is edited in two or more positions during the same revision, each block is presented to the network separately using the assumption that the edit blocks are independent, i.e., represent distinct edits. For the misclassifications however it can be the case that both blocks contribute to the same edit intention, e.g., rephrasing by deleting a sentence in one part of the page and adding a sentence somewhere else. When processing the blocks separately the network is unable to differentiate such a case from a simplifying or clarification revision, which explains the relatively high number of confusions between these classes. In the baseline decision tree classifier this aspect is incorporated as the *Number of Blocks* feature, which is indeed selected by the C4.5 algorithm. In future work this information could be incorporated in the LSTM classifier as well. In future work the treatment of revisions for which multiple categories apply can be investigated, for example by adding combination categories or by providing a ranking instead of a single class. However, this situation does not occur often, as most revisions clearly belong to only one category. Other improvements to the input representation can be obtained by incorporating the remaining input for longer revisions, which are currently cut off at 100 words. However, this information should be incorporated in a way that prevents sparsity in the input. Finally, the annotated corpus needs to be

extended in future work, in order to generalize the results and prevent overfitting the classifier.

The current experiments in general show promising results. The LSTM network classifies the 11 revision categories with approximately 70% accuracy, showing good classification performance across all categories. This performance is competitive compared to related research, despite the small training corpus in the current experiments and the observed overfitting on training data. The error analysis shows that confusions are generally explainable and that several types of errors may be addressed by extending the representation of the input. The current results are therefore a useful starting point for investigating the broader research question of automatic timeline extraction using relevant revisions. For this broader task several applications can be envisioned, in for example journalism or law enforcement. Although the process of automatic revision classification requires further improvements, the current research has shown to be able to contribute towards this task.

Acknowledgment

Author Marijn Schraagen is partly funded by the Dutch National Police Lab AI, https://icai.ai/police-lab-ai/.

References

1. Adler, T.B., de Alfaro, L., Mola-Velasco, S., Rosso, P., West, A.: Wikipedia vandalism detection: Combining natural language, metadata, and reputation features. Lecture Notes in Computer Science: Computational Linguistics and Intelligent Text Processing **6609**, 277–288 (2011)
2. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics **5**, 135–146 (2017)
3. Boyd, A.: Using Wikipedia edits in low resource grammatical error correction. In: Proceedings of the 2018 EMNLP Workshop W-NUT: The 4th Workshop on Noisy User-generated Text. pp. 79–84 (2018)
4. Daxenberger, J., Gurevych, I.: A corpus-based study of edit categories in featured and non-featured Wikipedia articles. In: Proceedings of COLING 2012: Technical Papers. pp. 711–726 (2012)
5. Daxenberger, J., Gurevych, I.: Automatically classifying edit categories in Wikipedia revisions. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. pp. 578–589 (2013)
6. Max, A., Wisniewski, G.: Mining naturally-occurring corrections and paraphrases from Wikipedia's revision history. In: Proceedings of LREC 2010. pp. 3143–3148 (2010)
7. Nelken, R., Yamangil, E.: Mining Wikipedia's article revision history for training computational linguistics algorithms. In: Proceedings of the AAAI Workshop on Wikipedia and Artificial Intelligence: An Evolving Synergy. pp. 31–36 (2008)
8. Yang, D., Halfaker, A., Kraut, R., Hovy, E.: Identifying semantic edit intentions from revisions in Wikipedia. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. pp. 2000–2010 (2017)