Chapter 5
A Survey of NLP Methods and Resources for Analyzing the Collaborative Writing Process in Wikipedia*

Oliver Ferschke, Johannes Daxenberger and Iryna Gurevych

Abstract With the rise of the Web 2.0, participatory and collaborative content production have largely replaced the traditional ways of information sharing and have created the novel genre of collaboratively constructed language resources. A vast untapped potential lies in the dynamic aspects of these resources, which cannot be unleashed with traditional methods designed for static corpora. In this chapter, we focus on Wikipedia as the most prominent instance of collaboratively constructed language resources. In particular, we discuss the significance of Wikipedia's revision history for applications in Natural Language Processing (NLP) and the unique prospects of the user discussions, a new resource that has just begun to be mined. While the body of research on processing Wikipedia's revision history is dominated by works that use the revision data as the basis for practical applications such as spelling correction or vandalism detection, most of the work focused on user discussions uses NLP for analyzing and understanding the data itself.

Oliver Ferschke
Ubiquitous Knowledge Processing Lab, Technische Universität Darmstadt,
e-mail: ferschke@ukp.informatik.tu-darmstadt.de

Johannes Daxenberger
Ubiquitous Knowledge Processing Lab, Technische Universität Darmstadt,
e-mail: daxenberger@ukp.informatik.tu-darmstadt.de

Iryna Gurevych
Ubiquitous Knowledge Processing Lab, Technische Universität Darmstadt,
German Institute for Educational Research and Educational Information,
e-mail: gurevych@ukp.informatik.tu-darmstadt.de

* This is a preprint version. The original publication will appear in the edited volume "The People's Web Meets NLP: Collaboratively Constructed Language Resources" in the Springer book series "Theory and Applications of Natural Language Processing" and will be available at www.springerlink.com.

5.1 Introduction

Over the past decade, the paradigm of information sharing in the web has shifted towards participatory and collaborative content production. In the early days of the Internet, web content has primarily been created by individuals and then shared with the public. Today, online texts are increasingly created collaboratively by multiple authors and are iteratively revised by the community.

When researchers first conducted surveys with professional writers in the 1980s, they found not only that the majority of them write collaboratively, but also that the collaborative writing process differs considerably from the way individual writing is done [25]. In collaborative writing, the writers have to externalize processes that are otherwise not made explicit, like the planning and the organization of the text. The authors have to communicate *how* the text should be written and *what* exactly it should contain.

Today, many tools are available that support collaborative writing for different audiences and applications, like *EtherPad*[2], *Google Docs*[3], *Zoho Writer*[4] or *BookType*[5]. A tool that has particularly taken hold is the *wiki*, a web-based, asynchronous co-authoring tool, which combines the characteristics of traditional web media, like email, forums, and chats [7]. Wiki pages are structured with lightweight *markup* that is translated into *HTML* by the wiki system. The markup is restricted to a small set of keywords, which lowers the entry threshold for new users and reduces the barrier to participation. Furthermore, many wiki systems offer visual editors that automatically produce the desired page layout without having to know the markup language. A unique characteristic of wikis is the automatic documentation of the revision history which keeps track of every change that is made to a wiki page. With this information, it is possible to reconstruct the writing process from the beginning to the end. Additionally, many wikis offer their users a communication platform, the *Talk pages*, where they can discuss the ongoing writing process with other users.

The most prominent example of a successful, large-scale wiki is *Wikipedia*, a collaboratively created online encyclopedia, which has grown considerably since its launch in 2001, and which contains over 22 million articles in 285 languages and dialects, as of April 2012. In this chapter, we review recent work from the area of Natural Language Processing (NLP) and related fields that aim at processing Wikipedia. In contrast to Medelyan et al. [19], who provide a comprehensive survey of methods to mine lexical semantic knowledge from a static snapshot of Wikipedia articles, we concentrate on the dynamic aspects of this resource. In particular, we discuss the significance of Wikipedia's revision history for applications in NLP and the unique prospects of the user discussions, a new resource that has just begun to be mined. Figure 5.1 gives an overview of the topics covered in this chapter. While the body of research on processing Wikipedia's revision history is dominated by works

[2] `http://etherpad.org/`

[3] `https://docs.google.com`

[4] `https://writer.zoho.com`

[5] `http://www.sourcefabric.org/en/booktype/`

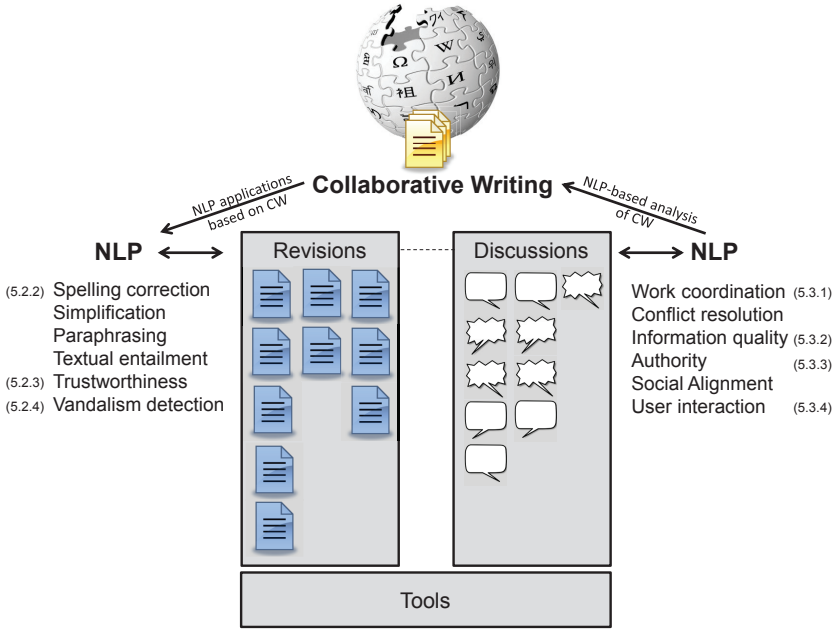


Fig. 5.1 The role of NLP in collaborative writing (CW): Topics covered in this chapter

that use the revision data as the basis for practical applications such as spelling correction or vandalism detection, most of the work focused on user discussions uses NLP for analyzing and understanding the data itself. Furthermore, there are increasing efforts to build tools and resources for enabling research on Wikipedia, which are discussed in the final section of this chapter.

5.2 Revisions in Wikipedia

Wikipedia's revision history lets us track the collaborative writing process in every single page in the encyclopedia. This section will explain the concept of revisions in Wikipedia and their uses for research in computational linguistics. After a short introduction to the concept of revisions in Wikipedia, we describe different NLP tasks that can benefit from the enormous data resulting from storing each single version of an article. Furthermore, we analyze applications of information coming from revisions with respect to article quality and trustworthiness. This is of general interest to computational linguistics, as the concepts and methods used can be applied to other collaboratively constructed discourse that uses revisions, in particular, wiki-based platforms.

5.2.1 The Concept of Revisions in Wikipedia

Throughout this chapter, we will use the term *page* to refer to a document in Wikipedia from any namespace, including articles, stubs, redirects, disambiguation pages, etc. The Wikipedia *namespace* system classifies pages into categories like Article or Talk, see Table 5.5. An *article* is a page from the Main namespace, usually displaying encyclopedic content. We call the Wikipedian who creates a new or edits an existing page its *author*. By storing his or her changes, a new revision of the edited page will be created. We call any version of a Wikipedia page a *revision*, denoted as r_v. v is a number between 0 and n, r_0 is the first and r_n the present version of the page, revisions are chronologically ordered. Registered authors can be identified by their user name, unregistered authors by the IP of the machine they are editing from. Wikipedia stores all textual changes of all authors for each of its pages. This way, it is possible to detect invalid or vandalistic changes, but also to trace the process of evolution of an article. Changes can be reverted. A *revert* is a special action carried out by users to restore a previous state of a page. Effectively, that means that one or more changes by previous editors are undone, mostly due to Vandalism (see Sect. 5.2.4). Authors can revert the latest page version to any past state or edit it in any way they wish.[6] A revert will also result in a new revision of the reverted page.

The *revision history* of a page shows every revision of that page with a timestamp (date and time of creation), the author, an optional flag for minor changes applied by the author, the size of the changes in bytes and an optional comment given by the author. We call these items revision *meta data*, as opposed to the textual content of each article revision. Having copies of each revision of a page, the changes between pairs of revisions can easily be accessed through Wikipedia's web page by so called diff pages. *Diff pages* display a line-based comparison of the wiki markup text of two revisions (see Fig. 5.2). In particular, the diff page for a pair of chronologically adjacent revisions r_v and r_{v-1} reflects the editing activity of one author at a certain point of time in the history of a page. We call the set of all changes from one revision to another a *diff*. A single diff in an article's revision history can be reverted if subsequent changes do not conflict with it, i.e. modify text affected by the reverted diff. This special kind of revert is usually referred to as *undo*.[7] As changes can affect one or several parts of a page, a diff can consist of various *edits*. An *edit* is a coherent local change, usually perceived by a human reader as one single editing action. In Fig. 5.2, two consecutive revisions r_v and r_{v-1} are displayed in a diff page, consisting of two edits inserting internal links. With respect to the meta data, the revisions in Fig. 5.2 have different authors. Both r_v and r_{v-1} are accompanied by comments. The timestamps indicate that the two versions have a time difference of approximately nine days.

---

[6] However, pages can be protected from editing by privileged users, as stated in the Wikipedia Protection Policy, see `http://en.wikipedia.org/wiki/WP:Protection_policy`.

[7] `http://en.wikipedia.org/wiki/WP:UNDO`

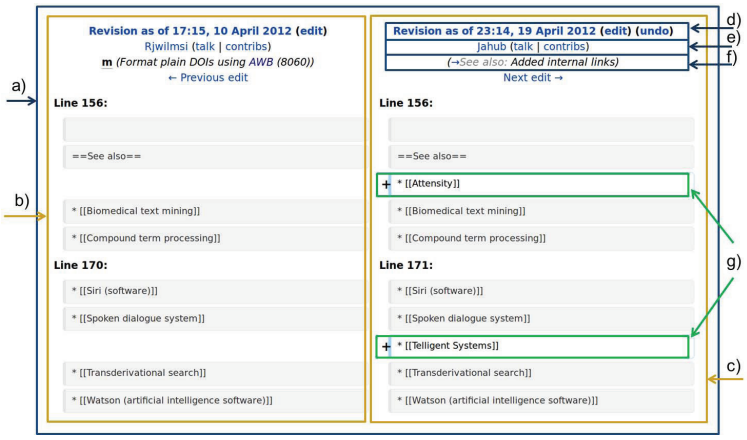


Fig. 5.2 A diff page: *a)* entire diff, *b)* older revision r_{v-1}, only the changed part and its context are displayed, *c)* newer revision r_v *d)* timestamp with edit and revert button, *e)* author, *f)* comment, *g)* edits. *d)*, *e)* and *f)* are meta data of r_v

The remainder of this section will discuss applications of the information that is encoded in revisions and how it is used as resource in NLP research. A list of tools to access revision data in Wikipedia can be found in Sect. 5.4.1.

5.2.2 NLP Applications

This section explains how computational linguistics can benefit from analyzing the revisions in collaboratively created discourse. We will present different approaches that are based on data from the Wikipedia revision history. These can be divided into three groups: *error detection*, *simplification* and *paraphrasing*. All of them benefit from the abundance of human-produced, near-parallel data in the Wikipedia revision history, as they employ it to extract task-specific training corpora on demand. See Table 5.1 for an overview.

In one of the first approaches to exploiting Wikipedia's revision history, Nelken and Yamangil [23] mine the English Wikipedia revision history to obtain training data for the detection of lexical errors, sentence compression, and text summarization. They apply different extraction algorithms on various levels of granularity, starting with the lexical level, to the sentence level, until the document level. The authors extract their data from a subset of the July 2006 English Wikipedia dump. A *dump* is a static snapshot of the contents of Wikipedia which may include all page revisions; for details see Sect. 5.4.2.

On the lexical level, they concentrated on a special type of error called eggcorns. Eggcorns are lexical errors due to both semantic and phonetic similarity, e.g. eggcorn is itself an eggcorn of the word acorn. The authors searched for cases of

word corrections in consecutive revisions r_v and r_{v-1} where corresponding words have been changed in such a manner that they are phonetically similar, but not morphologically related or synonyms. Since the semantic similarity in eggcorns is not well defined and thus hard to detect, they focused on detecting the phonetic similarity using the Editex algorithm [46]. A reference list of eggcorns, based on the so-called Eggcorn Database[8] which contains misspelled and correct word forms, serves to limit the article search space to those documents containing correct forms of one of the reference examples. For these articles, the authors harvested the revision history for pairs of revisions where r_v contains the correct form of an eggcorn. In the next step, they calculated edit distances between r_v and r_{v-1}, first, to identify similar sentences, and second, to find similar words within sentence pairs. Finally, the phonetic similarity of word pairs is measured. As for the resulting data, the authors report low precision, i.e. many false positives like typos or profanity. They justify that with their main goal to optimize the recall.

In another approach, Yamangil and Nelken [39] focus on the problem of data sparsity for applying a noisy channel model to sentence compression. First, they measure the edit distance between all sentences from pairs of adjacent revisions to find related sentences. In the resulting pairs of sentences, the authors look for edits adding or dropping words. That way, they detect sentence pairs being compressions of one another, assuming that all such edits retain the core meaning of the sentence. They verify the validity of the syntax of the extracted examples with a statistical parser, resulting in 380,000 parsed sentence pairs. This data is used within a syntax-based noisy channel compression model, which is based on an approach to sentence compression by Knight and Marcu [14]. In this model, a short sentence s is ranked by a source language model $p(s)$ and expands to a long sentence l with a certain probability $p(l|s)$. In [14], $p(l|s)$ corresponds to the probability of the syntax tree of l to be transformed into the syntax tree of s. For a long sentence l, the model seeks the short sentence s that is most likely to have generated l, that is to maximize $p(s) \cdot p(l|s)$. Yamangil and Nelken's model benefits from the mass of training data

Table 5.1 Previous approaches using the Wikipedia revision history as source of training data

Reference	Type of data	Size of labeled data	Language	Publicly avail.
[23]	eggcorns (malapropisms)	108[a]	English	no
[39]	sentence compression pairs	380,000	English	no
[43]	real-word spelling error pairs	686	English, German	yes
[18]	spelling errors and paraphrases	146,593[b]	French	yes
[40]	lexical simplifications pairs	4049[c]	English	yes
[38]	lexical simplifications pairs	14,831	English	no
[41]	textual entailment pairs	1614	English	yes

[a] This number is based on a statement in [23] saying that they "successfully found 31% of the reference eggcorns", the latter summing up to 348.
[b] Refers to the spelling error corpus, v2.0.
[c] Sum of pairs from the edit model and the meta data model.

[8] `http://eggcorns.lascribe.net/`

as it offers enough examples to add lexical information to the syntactic model. The model thereby learns probabilities not only based on the syntax trees of the example sentence pairs, but based on their words. Their result shows an improvement in compression rate and grammaticality over the approach of Knight and Marcu. However, they experience a slight decrease in the importance of the resulting sentences. *Importance* measures the quality of information preservation in the compressed version of a sentence with regard to the original. The authors explain this drop with the training data originally coming from both compressions and expansions (i.e. decompressions) by authors in Wikipedia, where the latter seems to frequently add important information that should not be skipped.

Nelken and Yamangil [23] also present a method for summarization of whole articles or texts. It is based on the assumption that a sentence with high persistence throughout the edit history of an article is of significant importance for it, i.e. it can be used for summarization purposes. They define a weak sentence identity, which allows for small changes in persistent sentences and is defined by a threshold of the edit distance. The authors tested the usability of their approach on two Wikipedia articles and found that the first sentence of a section as well as structural markup (such as link collections) have a higher persistence. As Nelken and Yamangil state, their methods cannot replace a full summarization application, but would be useful as part of a larger system.

In conclusion, Nelken and Yamangil present a series of promising applications for data extracted from the Wikipedia revision history. Their proposals for error detection, sentence compression and text summarization lay a foundation for further approaches working with this kind of data. Advanced systems can benefit from Nelken and Yamangil's insights. A first step would be to normalize data extraction from revision history, e.g. to classify any edits between a pair of revisions into categories like vandalism, spelling error corrections or reformulations before they are further processed. An automatic classification of edits facilitates the approaches building upon revision history data and might help to increase the precision of such systems. This holds not only for the approaches outlined by Nelken and Yamangil, but also for most of the applications presented in the remainder of this section. We roughly divided them into the domains of spelling error correction and paraphrasing.

Spelling Error Correction

Zesch [43] extracts a corpus of real-word spelling errors (malapropisms), which can only be detected by evaluating the context they appear in. For example, in the sentence, "That is the very defect of the matter, sir" (from Shakespeare's "The Merchant of Venice"), *defect* is confused with *effect*. This type of error is generally not detected by conventional spelling correctors. In the past, training and/or test data for this type of error has mostly been created artificially, e.g. by automatically replacing words with similar words from a dictionary. Zesch's approach is an attempt to generate a corpus of naturally occurring malapropisms. Therefore, the author extracts pairs of sentences with minimal changes from consecutive Wikipedia revi-

sions. To determine such sentence pairs, a restrictive filter is applied on each pair of phrases in adjacent revisions, ruling out pairs that are either equal or exceeding a small threshold in their character length difference. The sentences are annotated with part-of-speech tags and lemmata. Further filters ensure that the sentences differ in just one token, which may not be a number or a case change. The edit distance between the old and the new token must be below a threshold. With regard to the semantic level, edits involving misspelled words (the edit must not be an error that can be detected using a conventional spelling corrector), stopwords, named entities (the new token may not be a named entity) and semantically motivated changes (direct semantic relations are determined using WordNet) are filtered out. The resulting dataset was manually corrected to remove cases of vandalism and examples that could not be ruled out by the above described filter mechanism because they did not provide enough context. It has been generated from five million English and German revisions and contains altogether 686 error pairs.[9] The author used his data to compare statistical and knowledge-based approaches for detecting real-word spelling errors. Through this analysis he shows that artificial datasets tend to overestimate the performance of statistical approaches while underestimating the results of knowledge-based ones. This way, he proves the usefulness of the corpus of naturally occurring real-word errors created from the Wikipedia revision history, because it offers a more realistic scenario for the task of evaluating real-word spelling error correction.

In another application working with spelling errors, Max and Wisniewski [18] present the Wikipedia Correction and Paraphrase Corpus (WiCoPaCo). Different to the aforementioned approach, they analyze various types of edits. Their data originates from article revisions in the French Wikipedia. Differences between modified paragraphs from adjacent revisions are determined using the longest common subsequence algorithm. Only edits with a maximum of seven changed words are kept. Edits which exclusively add or delete tokens are not considered. Further filters rule out edits changing more than a certain number of words, changes that only affect punctuation and bot edits. *Bots* are automatic scripts operating in Wikipedia to carry out repetitive tasks, mostly for maintenance. The remaining data is tokenized and markup is removed. The actual edit is aligned in the context of the paragraphs of r_v (denoted by </before>) and r_{v-1} (</after>), see Fig. 5.3 for an example. The resulting corpus consists of 408,816 edits (v2.0), coded in an XML format as shown in Fig. 5.3. This format stores, together with the textual data, meta data such as the user comment (denoted by `wp_comment`). To build a corpus of spelling errors, the authors filter out 74,100 real-word errors and 72,493 non-word errors using a rule-based approach. Among all edits affecting only a single word, they apply two rules. First, a `hunspell` correction system detects for both r_{v-1} and r_v whether the modified word w_v or w_{v-1} is in the dictionary. This way, they find:

- non-word corrections (w_{v-1} is erroneous, w_v is correct),
- real-word errors and paraphrases (both w_v and w_{v-1} are correct)

[9] Freely accessible at `http://code.google.com/p/dkpro-spelling-asl/`.

```
<modif id="142" wp_before_rev_id="1842309" wp_after_rev_id="1842337"
    wp_comment="Statistiques">
  <before>
    Taux de croissance de la <m num_words="1">pop.</m>: 0,24% (en 2001)
  </before>
  <after>
    Taux de croissance de la <m num_words="1">population</m>: 0,24% (en 2001)
  </after>
</modif>
```

Fig. 5.3 A slightly truncated entry from the WiCoPaCo, the coded edit in the </m>-tag is highlighted

- and proper noun or foreign word edits, spam and wrong error corrections (w_v is erroneous).

The second rule distinguishes between real-word errors and paraphrases. Therefore, a maximum character edit distance of 3 between w_v and w_{v-1} is allowed for spelling corrections, assuming that most of the spelling error corrections change 3 or less characters. Additionally, for the non-word corrections, edits with a character edit distance greater than 5 are ruled out. The authors justify this step with the need to filter out spam. They published the resulting data as a freely available corpus, the WiCoPaCo.[10] To evaluate the spelling error subset of WiCoPaCo, the authors randomly split the data into training and test set. They create candidate sets based on both `hunspell` rules and error correction patterns from their corpus. The later comprises two lists:

- a list of words built by applying the most frequent error correction scripts (e.g. e $\rightarrow$ é) extracted from their corpus to misspelled words and
- a list with all corrections of misspelled words from the training set.

For evaluation, they count the number of candidate sets containing the correct word, using the training set to build the candidate sets. The results show that the combined approach improves over a system based solely on `hunspell`. Improvement is particularly high for real-word errors. This is in line with the findings by Zesch [43] who also pointed out the importance of naturally occurring real-word error datasets. However, Max and Wisniewski do not test their approach on different data than the WiCoPaCo corpus.

After manual inspection of the WiCoPaCo data, Max and Wisniewski also developed a classification system to categorize edits in Wikipedia. Their system separates changes which preserve the meaning from those that alter the meaning. The former are further divided into edits modifying the spelling (such as spelling errors) and edits modifying the wording (e.g. paraphrases). Edits altering the meaning are divided into spam and valid meaning changes (such as simplifications). Based on the paraphrases in their corpus, the authors analyzed the probabilities of transformations of POS sequences (e.g. DET ADJ NOM $\rightarrow$ DET NOM). As a possible application,

[10] See `http://wicopaco.limsi.fr/`.

they propose employing these probabilities to assess the grammaticality of paraphrases when several candidates exist. The quantitative analysis and classification of paraphrases in WiCoPaCo is subject to future work.

Paraphrasing

Yatskar et al. [40] present an unsupervised method to extract lexical simplifications in the Simple English Wikipedia. They do not aim at simplifying entire sentences, but words or expressions, e.g. when "annually" is replaced by the simpler version "every year". In order to obtain a training corpus, they extract sentence pairs from adjacent revisions. Alignment of sentences is carried out based on the cosine similarity measure utilizing TF-IDF scores [22]. To calculate the latter, sentences are treated as documents and adjacent revisions as the document collection. From aligned sentences, the longest differing segments are calculated (*edits*) and changes longer than five words are filtered out. The authors introduce two different approaches to extract simplifications.

In the first approach (*edit model*), probabilities for edits to be simplifications derive from a model of different edits that are performed in the Simple and the Complex English Wikipedia. Based on edits in the Simple Wikipedia, the probability for an edit to be a simplification is calculated. On the opposite, the Complex English Wikipedia is used to filter out non-simplifications. To do so, the authors make the simplifying assumption that all edits in the Complex Wikipedia correspond to what they call "fixes", i.e. spam removal or corrections of grammar or factual content. Furthermore, they assume that vandalism does not exist and that the probability of a fix operation in the Simple Wikipedia is proportional to the probability of a fix operation in the Complex English Wikipedia. In their second approach (*meta data model*), Yatskar et al. use revision meta data to detect simplifications, namely the revision comments. They inspect all revisions containing the string "simpl" in their comment. Among the detected revisions, all possible edits are ranked by an association metric (Pointwise Mutual Information).

In a preliminary evaluation, the top 100 sentence pairs from each approach and a random selection from a user-generated list[11] have manually been annotated by native and non-native speakers of English as being a simplification or not. The inter-annotator agreement among the three annotators is sufficient with $\kappa = 0.69$. The authors used baselines returning the most frequent edits and random edits from the Simple English Wikipedia; both yielded a precision of 0.17. For the meta data method, a precision of 0.66 is reported, the edit model approach achieved a precision of 0.77, whereas the user-generated list had the highest precision with 0.86. With regard to recall, the authors report that the edit model generated 1,079 pairs and the meta data model 2,970 pairs, of which 62% and 71% respectively, were not

[11] The Simple Wikipedia author `Specerk` offers a list of transformation pairs: `http://simple.wikipedia.org/w/index.php?title=User:Spencerk/list_of_straight-up_substitutables`.

included in the user-generated list. The annotated datasets have been published and are freely available.[12]

In a similar approach, Woodsend and Lapata [38] additionally use syntactic information for a data-driven model of sentence simplification. Like Yatskar et al., they obtain their training data from the Simple and the Complex English Wikipedia. Two methods to create parallel corpora of simple and complex sentences are applied: first, the authors align sentences from Simple and Complex English Wikipedia articles (article corpus), and second, they align sentences from adjacent revisions in the Simple Wikipedia (revision corpus). In both corpora, the markup is removed. In the article corpus, the authors align parallel articles via the interwiki (language) links between the Simple and the Complex Wikipedia. Sentence alignment in parallel articles is established using TF-IDF scores to measure sentence similarity [22]. In the revision corpus, they select suitable revisions according to comment keywords, e.g. "simple", "clarification" or "grammar". Appropriate revisions r_v are compared to r_{v-1} followed by calculating the diff to find modified sections. Within those, the corresponding sentences are aligned via a word-based diff, resulting in 14,831 paired sentences. The aligned sentences are syntactically parsed. The parsed sentence pairs are used to train a Quasi-synchronous grammar (QG, similar to the content-based method of Zanzotti and Pennacchiotti [41], cf. below). Given a syntax tree T_1, the QG generates monolingual translations T_2 of this tree. Nodes in T_2 are aligned to one or more nodes in T_1. Alignment between direct parent nodes takes place when more than one child node (lexical nodes, i.e. words) are aligned. This way, a set of lexical and syntactic simplification rules as well as sentence splitting rules are generated, yielding transformations such as the following, which splits a sentence:

> John Smith walked his dog and afterwards met Mary. →
> John Smith walked his dog. He met Mary later.

Woodsend and Lapata solve the problem of finding the optimal QG transformations to simplify source sentences with an integer linear programming approach. In short, they use an objective function which guides the transformation towards a simpler language of the output, e.g. a lower number of syllables per word or of words per sentence. The authors evaluate their approach based on human judgments and readability measures. Human judgments include an evaluation of the output sentence with respect to the readability (whether is was easier to read than the input sentence), the grammaticality and the preservation of the meaning. The models are tested on the dataset used in Zhu et al. [45], who also align sentences from the Simple and the Complex English Wikipedia. With regard to the calculated readability measures, both the model trained on the revision corpus and the model trained on the article corpus do not outperform a baseline relying on the user-generated list previously used by Yatskar et al. [40] (cf. footnote 11). Considering the human judgments, the model trained on the revision corpus outperforms the article corpus model in all of the evaluation aspects. This result supports our assumption that the incorporation of revision history data not only helps to increase the amount of training data but also improves the performance of certain NLP applications.

[12] See `http://www.cs.cornell.edu/home/llee/data/simple/`.

Zanzotti and Pennacchiotti [41] apply semi-supervised machine learning for the task of Recognizing Textual Entailment (RTE) pairs from the Wikipedia revision history. They describe four essential properties of a textual entailment dataset and why data coming from Wikipedia's revision history is appropriate for this, i.e. the data is

- *not artificial*, as it is extracted from authentic Wikipedia texts
- *balanced*, i.e. equal in number of positive entailment pairs (when new information is added to the old content or old content is paraphrased) and negative entailment pairs (when the new information contradicts the old content or the entailment is reverse); this is roughly the case for their data as shown in the following
- *not biased with respect to lexical overlap*, i.e. the lexical overlap of positive and negative entailment pairs should be balanced, this is mostly true for the Wikipedia revision data, as usually only a few words are changed both for positive and for negative entailment pairs
- *homogeneous to existing RTE corpora* with respect to the entailment pairs contained in these corpora, this is roughly the case for their data as shown in the following.

Their approach to separate positive lexical entailment candidates from negative ones is based on co-training. Co-training is designed to learn from labeled data L and unlabeled data U and has to access the corpus in two different and independent views. Two different classifiers, each of them working with features from one of the two views, are trained on copies of L, defined as L_1 and L_2. These classifiers are used to classify data from U, resulting in different classifications U_1 and U_2. Finally, the best-classified examples in U_1 are added to L_2, resp. U_2 to L_1. This procedure is iteratively repeated until a stopping condition is met. As for the two views, the authors suggest a *content-based view* (features based on the textual difference of two revisions) and a *comment-based view* (features based on the comment of r_v). The features in the content-based view rely on syntactic transformations. A feature will be activated, if the syntactic transformation rule associated with that feature unifies with the syntax tree representations of a pair of sentences. The authors do not specify in detail how these pairs are generated. In the comment view, features are based on a bag-of-words model. The latter is calculated from the comment words, which have been filtered with respect to the stop words.

For the evaluation of their approach, Zanzotto and Pennacchiotti randomly selected 3,000 instances of positive and negative entailment pairs from 40,000 English Wikipedia pages (*wiki_unlabeled* dataset). Additionally, they manually annotate 2,000 entailment pairs. The inter-annotator agreement on a smaller development corpus of 200 examples is $\kappa = 0.60$. After removing vandalism and spelling corrections they obtained 945 positive and 669 negative entailment pairs (*wiki* dataset). The datasets are freely available.[13] The authors compared the *wiki* dataset to other corpora from RTE Challenges, namely the datasets from the RTE-1, RTE-2 and RTE-3 challenges [10]. To evaluate the quality of the *wiki* dataset, they split it into

[13] See `http://art.uniroma2.it/zanzotto/`.

equally sized development, training and test set. The classification of positive and negative entailment pairs is carried out by a Support Vector Machine trained on the features from the content-based view. The authors report an accuracy of 0.71 for that approach when applied to the *wiki* data, compared to 0.61 for the RTE-2 dataset. Combining *wiki* data with the RTE challenge datasets for training did not show significant decrease or increase of accuracy. Therefore, the authors conclude that the *wiki* dataset is homogeneous to the RTE datasets. To evaluate the co-training approach, they use RTE-2 as labeled set and *wiki_unlabeled* as unlabeled set. RTE-2 does not allow for the comment-based view. Hence, the comment-view classifier is not activated until the first training examples are added from the content-based classifier. Performance is reported to become stable after several iterations with approximately 40 unlabeled examples and accuracy around 0.61. The authors conclude that their semi-supervised approach successfully serves to expand existing RTE datasets with data extracted from Wikipedia.

Having discussed example NLP applications based on the Wikipedia revision history data, we now focus on how revision information can be used to assess article quality.

5.2.3 Article Trustworthiness, Quality and Evolution

The revision history of a page in Wikipedia contains information about how, when and by whom an article has been edited. This property has been used to automatically assess the quality of an article. In this context, quality in Wikipedia is related to trustworthiness. However, the trustworthiness of an article and its quality are not necessarily the same. Rather, trustworthiness can be seen as a means of successfully communicating text quality to users. In the context of Wikipedia, trustworthiness is often related with the skills and expert knowledge of the author of a revision, whereas quality is rather measured in terms of the textual content itself. Certainly, this distinction is not always made, and different studies use the terms differently and sometimes interchangeable. In the following, we present several studies that make use of the Wikipedia revision history to analyze article quality and trustworthiness. An overview of the approaches can be found in Table 5.2.

Table 5.2 Trustworthiness and article quality assessment approaches based on the Wikipedia revision history

Reference	Type of Revision Features	Criteria for Evaluation	Language
[42]	author reputation, edit type	featured, cleanup, other	English
[5]	author score, edit size	featured, articles for deletion	Italian
[37]	quantitative surface features	featured, non-featured	English
[33]	semantic convergence	good, non-good	English
[11]	revision cycle patterns	featured, good, B-, C-class, start, stub	English

Zeng et al. [42] were one of the first to develop and evaluate a model of article trustworthiness based on revision histories. Their model is based on author reputation, edit type features and the trustworthiness of the previous revision. As for the edit type features, the number of deleted and/or inserted words is measured. The reputation of authors is approximated by their editing privileges. Certain actions in Wikipedia, e.g. blocking other users, can be carried out only by privileged users. Furthermore, registered authors can be distinguished from unregistered users and blocked users. The authors apply a Dynamic Bayesian network depending on these features to estimate the trustworthiness of a revision based on a sequence of previous states, i.e. revisions. To account for uncertainty in the trustworthiness of authors and in the edit type features, beta probability distributions for the trustworthiness values of the network are assumed. The trustworthiness of r_v is equal to the trustworthiness of r_{v-1} plus the inserted trustworthy content minus the deleted trustworthy content, i.e. incorrectly removed portions of text. The amount of trustworthy and untrustworthy content is determined by an author's reputation. To evaluate the model, the authors built a corpus of internally reviewed articles, altogether containing 40,450 revisions. Wikipedia has an internal review system which labels articles that meet certain predefined quality[14] criteria, e.g. they should be comprehensive, contain images where appropriate, etc. The highest rating of an article is *featured*. However, distinguished articles not yet fulfilling all criteria to be featured can be also labeled as *good*. On the contrary, articles tagged for *cleanup*, do not meet the necessary quality standards as defined in the Wikipedia Manual of Style.[15] To evaluate their model, Zeng et al. calculate an article's mean trust distribution, an indicator of the trustworthiness of its latest revision, based on the above Bayesian network. They find that featured articles have the highest average of mean trust distributions, while cleanup articles show the lowest values. The authors carry out a manual inspection of changes in average trustworthiness values throughout the history of an article, showing that these changes correspond to major edit types like insertions or deletions of large quantities of text. The model is thus able to reproduce a realistic picture of the trustworthiness of articles based on their revision history.

Cusinato et al. [5] have a similar view on article quality in Wikipedia, as proposed in their system called QuWi. The system is based on an approach originally developed for quality assessment in peer reviewed scholarly publishing introduced by Mizzaro [21]. This model assigns quality and steadiness scores to articles, authors and readers. Scores for each of them are updated when a reader judges a paper, based on the following assumptions: the scores of authors are bound to and updated with the judgment scores of their articles, weighted with the article's steadiness. The weight of an article judgment depends on the score of the reader rating it. Readers' scores are bound to and updated with the appropriateness of their judgments, based on their agreement with the average rating of the articles they judged. Steadiness for articles, authors and readers increases with every corresponding judgment made. To adapt this model to Wikipedia, Cusinato et al. use the following adjustments:

[14] `http://en.wikipedia.org/wiki/WP:FA_Criteria`

[15] `http://en.wikipedia.org/wiki/WP:Manual_of_Style`

1. Articles have more than one author, hence, judgments have to be based on single contributions, i.e. edits between adjacent revisions.
2. Edits cannot be rated directly, hence, the readers' judgment on an edit is measured implicitly by analyzing the next edit on the article, i.e. the next revision. In other words, the author of r_v automatically becomes the reader of r_{v-1}.

Modifications express negative votes, while unmodified content is considered to be positive. The score of a contribution is calculated based on the ratio between modified and unmodified text (i.e. the reader's judgment), weighted by the score of the reader. Based on contribution scores, author and reader scores are calculated as explained above, derived from the approach by [21]. Finally, article scores are assigned based on the scores of the words contained in the article, weighted by the word length. Word scores are calculated based on the author's and (previous) readers' scores, each of them averaged by their steadiness scores.

The authors tested their system on 19,917 articles from the Science category of the June 2007 snapshot from the Italian Wikipedia. They ran the score calculation on the entire set of revisions and recorded article scores at six equally distributed timestamps, including the latest ones. As expected, average article scores increase over time. Featured articles and articles proposed for deletion were used to evaluate the calculated scores. The average score (ranging between 0 and 1) of the 19 featured articles contained in their corpus is 0.88, significantly higher than the total average 0.42, whereas 75 articles for deletion have an average score of 0.27. This work demonstrates an interesting way to apply Wikipedia revision information to an existing model of quality assessment and has been shown to work successfully on a small part of the Italian Wikipedia revision history. The approach could further be improved by accounting for bot edits and vandalism.

Wilkinson and Huberman [37] analyze correlations between quantitative revision features and article quality. Based on the number of revisions made in all Wikipedia articles they develop a model of article growth. In this model, the number of revisions in a given timeframe is on average proportional to the number of previous changes (i.e. revisions) made to the article. As a result, older articles are edited more often, or, as the authors put it: "edits beget edits". They verify their assumption with an empirical analysis over all page revisions in the English Wikipedia between January 2001 and November 2006, except for redirect and disambiguation pages and revisions by bots. Furthermore, they explore correlations between article quality and editing. Therefore, the authors analyze age- and topic-normalized featured and non-featured articles according to their number of revisions and their number of distinct authors. Their findings show that featured articles have a statistically significant higher number of revisions and distinct authors when compared to non-featured articles. To account for the cooperation among authors, they do the same for Talk pages (see Sect. 5.3), resulting in an even more significant difference between featured and non-featured articles. The authors conclude that high-quality articles in Wikipedia can be distinguished from other articles by the larger numbers of article edits, Talk pages edits and distinct authors.

Analysis of Article Lifecycles

Using the revision count as a proxy for article quality seems to yield interesting results. However, it must be considered that featured articles in Wikipedia receive special attention because of their status. Therefore, the following approaches go further and explicitly treat articles as constructs going through different phases or stages of maturity, i.e. they study the evolution. The revision history is the only source of information for this purpose.

Thomas and Sheth [33] introduce a notion of article stability, which they call Semantic Convergence. They assume an article to be mature (i.e. trustworthy), when it is semantically stable. Semantic stability is defined in terms of semantic distance in a TF-IDF vector space representation of revision milestones. The TF-IDF space is calculated over all words occurring in an article's entire revision history. A revision milestone is defined as a combination of all revisions in one week, word counts for milestones are calculated as medians. This way, the authors aim to balance different editing frequencies for individual articles. They test their hypothesis on a dataset of 1,393 articles labeled as good and 968 non-labeled articles with a revision history consisting of at least 50 revisions. For evaluation, they measure the pairwise cosine distance between adjacent revision milestones and the distance between every revision milestone and the final revision. They show that articles generally move towards a stable state, i.e. that the semantic distance between revision milestones drops with time. When it comes to predicting the maturity for a single article at a given point of time, their measure does not prove to be reliable. However, knowledge about the past of an article helps to detect its present state, because articles which have already undergone stable revision milestones are less likely to change. Good and non-good articles did not show a significant difference in terms of their stability. Hence, if an article is labeled as good, it does not necessarily mean that its content is stable.

Han et al. [11] use a Hidden Markov Model to analyze the history of a Wikipedia article as a sequence of states. They define a number of states an article usually passes before reaching a convergence state. The states in their Markov model are as follows: building structure, contributing text, discussing text, contributing structure and text, discussing structure and text/content agreement. The observation variables used to determine the Markov states are calculated between a pair of consecutive revisions and are divided into:

- update type (insertion, deletion, modification),
- content type (structure, content, format) and
- granularity type (extent of the edit).

Sequences or series of sequences of states are combined to form so called Revision Cycle Patterns. The authors aim to find correlations between human evaluated quality classes and revision cycle patterns to automatically assess the quality of an article. Therefore, they test their model on a corpus containing articles which have

been labeled according to the Wikipedia internal quality grading scheme[16] as either featured, A-class, good, B- and C-class as well as start and stub-class. They create a model based on the following steps. First, Revision Cycle Patterns for each quality class in the corpus are extracted. Recurring sequences of states are detected via frequent items mining. Second, these are clustered to discover the dominant patterns and third, clusters of cycle patterns are related with quality labels. With this method, the percentage of correctly classified articles is between 0.98 for featured articles and 0.85 for the stub class. The authors report that their approach outperforms the results in Dalip et al. [6], who work on the same task and data, but without using features based on revision history data. Thus, the revision history based features turn out to be helpful for this task.

A combination of the revision-related features with language features regarding style, structure or readability as presented in [6] is an emerging topic. To the best of our knowledge, no effort has yet been made to incorporate all of the available revision-based information with plain text language features to assess article quality. This indicates a promising direction for future work on quality assessment and trustworthiness. Furthermore, as already mentioned, a clear definition of quality and trustworthiness in Wikipedia has not been established yet. The above outlined studies all have slightly different concepts of quality and trustworthiness. The evaluation methods are almost exclusively based on the human assigned Wikipedia-internal quality labels as explained above. This is a shortcoming, as the criteria for these ratings can change over time, and the quality assessment process may not be reproducible for external raters. A broader analysis of article quality which goes beyond user-assigned labels, together with a comprehensive definition of text quality, is thus required.

5.2.4 Vandalism Detection

This subsection explains the usage of revision history data to detect spam or vandalistic edits in Wikipedia. Vandalism is a major problem in Wikipedia, since anybody can edit most of its content. About 6 to 7% of all revisions in the English Wikipedia are estimated to be vandalized [3, 26]. In short, vandalism or spam is "any addition, removal, or change of content in a deliberate attempt to compromise the integrity of Wikipedia".[17] Vandalistic additions, removals or changes to an article can only be detected using revision history data, because at least two revisions need to be compared: a trustworthy, not vandalized revision r_{v-1} and a possibly vandalized revision r_v. Malicious edits are supposed to be reverted as quickly as possible by other users, which in practice seems to work quite well. Different median survival times

[16] WikiProject article quality grading scheme: `http://en.wikipedia.org/wiki/Wikipedia:Version_1.0_Editorial_Team/Assessment`.

[17] From `http://en.wikipedia.org/w/index.php?title=Wikipedia:Vandalism&oldid=489137966`. The same page also offers a list of frequent types of vandalism.

Table 5.3 Vandalism detection approaches and the features they use. For each of them, the best classification results on two corpora are given. Please note that these numbers are not entirely comparable due to the use of different classifiers (given in brackets) and different training and test sets

Reference	Basic Feature Types[a]	WEBIS-VC07	PAN-WVC 10
[28]	T, L, M, R	0.85 F_1 (Log. Regr.)	–
[4]	statistical language model	0.90 F_1 (J48 Boost)	–
[36]	T, L/S, M, R	0.95 F_1 (Log. Regr. Boost)	0.85 F_1 (Log. Regr. Boost)
[1]	T, L, M, R	–	0.98 AUC (Rand. Forest)
[12]	T, L, M, R	–	0.97 AUC (Rand. Forest)

[a] T = Textual, L = Language, S = Syntax, M = Meta Data, R = Reputation

for vandalized revisions are quoted, ranging from less than three minutes [34] to 11.3 minutes [13], depending on the type of vandalism.

Wikipedia has a revert system which serves to undo unwanted edits (cf. Fig. 5.2 *d*) and particularly, vandalistic edits. A small number of automatic bots watch changes and revert obvious vandalism. At the time of writing, `ClueBot NG` was the main anti-vandalism bot in the English Wikipedia.[18] In contrast to many of the previous rule-based anti-vandalism bots, its detection algorithm is based on machine learning techniques.

The International Competition on Wikipedia Vandalism Detection is a good starting point for work on vandalism detection in Wikipedia. It evaluates vandalism detection based on the PAN Wikipedia vandalism corpus (WVC) 10 and 11.[19] Each of these corpora contains around 30,000 edits of Wikipedia articles labeled as *regular* or *vandalism*.

State-of-the-art approaches formulate Wikipedia vandalism detection as a machine learning task. Malicious edits have to be separated from regular ones based on different features. For an overview of the approaches, see Table 5.3. Vandalism detection approaches can be classified according to their adoption of features. We categorize the features as proposed in Adler et al. [1]:

- Textual Features: language independent
- Language Features: language specific
- Meta Data Features: author, comment and timestamp
- Reputation Features: author and article reputation

Although most of the presented studies use this kind of distinction between different types of features, there is no absolute agreement on how to categorize them. Likewise, some works might include an *author-is-registered* feature to meta data, while others consider such a feature as author reputation. Textual features

[18] Cf. a list of Anti-vandalism bots compiled by the author `Emijrp`: `http://en.wikipedia.org/w/index.php?title=User:Emijrp/Anti-vandalism_bot_census&oldid=482285684`.

[19] See `http://www.webis.de/research/corpora/pan-wvc-10` and `http://www.uni-weimar.de/cms/medien/webis/research/corpora/pan-wvc-11.html`.

Table 5.4 Examples of features for vandalism detection as used in [1]

Textual	Language	Meta Data	Reputation
Ratio digits to characters	Freq. of vulgarisms	Comment Length	Author behavior hist.
Ratio upper/lowercase characters	Freq. of pronouns	Author is registered	Author geogr. region
Length of longest token	Freq. of typos	Time of day	Reputation for article

involve language-independent characteristics of an edit, such as the number of changed characters, upper- to lowercase ratio and similar. To analyze (natural) language features, language-related knowledge is required. This knowledge comes from language-specific word lists or dictionaries (of swearwords etc.), and/or further processing with NLP tools (POS-tagger, parser etc.). Meta data features refer to information coming from a revision's meta data, like the comment or the time of its creation. Reputation features need detailed information about the author or edited article, e.g. the number of an author's past edits or the article's topical category. Examples of such features can be found in Table 5.4.

Potthast et al. [28] are among the first to present several important features for vandalism detection. They used the so called WEBIS-VC07 corpus which contains 940 pairs of adjacent revisions. Among them, the authors manually labeled 301 vandalistic revisions. Their features are mainly based on textual differences between revisions, including char- and word-based ones as well as dictionaries to detect vulgar words. They use some features based on meta-data. Among these, the *edits-per-user* feature shows the highest recall. The authors cross-validate a classifier based on Logistic Regression on their dataset, comparing the classifier's result to the baseline performance of rule-based anti-vandalism bots. They report 0.83 precision at 0.77 recall using a classifier based on logistic regression, outperforming the baseline primarily with respect to the recall. It should be noted, however, that anti-vandalism bots are designed to have high precision aiming to avoid reverting false positives, i.e. revisions that have not been vandalized. Insofar, vandalism detection approaches might focus on high precision rather than high recall.

Chin et al. [4] do not use any meta information in their vandalism classification approach. Their features come from a statistical language model, which assigns probabilities to the occurrence of word sequences. To account for repeated vandalism, they not only compare a revision to the immediately preceding one, but also to other preceding ones. Their test corpus consists of the entire revision history for two large and often vandalized articles. Instead of labeling huge amounts of data by hand, they apply an active learning approach, where classifiers are built iteratively, starting with the labeled data from the WEBIS-VC07 corpus [28]. After each iteration, only the 50 most probable vandalism edits are manually labeled. An evaluation using a Boosting classifier with J48 Decision Trees based on their features resulted in 0.90 F_1 on this corpus. The increase over the approach in Potthast et al. [28] is primarily due to a higher precision. Additionally, Chin et al. classify types of vandalism and edits in general. Consequently, they not only label the presence or absence of vandalism but also its type, e.g. Graffiti (inserting irrelevant or pro-

fane text), Large-scale Editing (inserting or replacing text with a large amount of malicious text) or Misinformation (replacing existing text with false information). Their analysis shows that Graffiti accounts for more than half of all vandalistic edits. The authors used a Logistic Regression classifier and Support Vector Machines as models for their active learning approach. After three to four iterations, the Logistic Regression classifier yielded best results with 0.81 average precision. An error analysis shows that the wiki markup and unknown words (e.g. template names) cause the language model to fail. The model considers unknown strings as out-of-vocabulary and hence assigns them a high probability of being vandalism. Furthermore, separating reverts from vandalized revisions turns out to be difficult. This could be addressed by including meta data like the comment into their features.

Wang et al. [36] focus on syntactic and semantic features in their approach of vandalism classification. They distinguish lexically ill-formed, syntactically ill-formed and ill-intentioned types of vandalism, aiming to account for cases of vandalistic edits or spam which are harder to recognize. This happens when authors try to hide bad intentions by inserting well-formed off-topic comments, biased opinions or (implicit) advertisement. To detect this type of edits, a classifier needs to have access to a wider range of language features. Therefore, the authors apply what they call shallow syntactic and semantic modeling. Their features are based on the title of an article and the diff text between adjacent revisions. Since this data contains sparse information for training a classifier, they use it as a query to various web search engines to build article-specific models. The top-ranked retrieved results are labeled with POS-tags. The authors use n-grams of POS-tags alone as well as n-grams of POS-tags and words to model a syntactic and a semantic representation of each revision. Their classifier also uses meta data (the comment) and reputation (number of revisions per author) features along with lexical ones (e.g. vulgarism and web slang). The experiments are conducted with the WEBIS-VC07 and the PAN-WVC 10, both split into training and test set. Classification is done by a Logistic Model Trees classifier and a Logistic Regression classifier with Boosting. With the WEBIS-VC07 corpus and the Logistic Regression Boosting classifier using all of their features they report the highest F_1 of 0.95. This is an improvement of around 15% compared to the results in [28]. On the PAN-WVC 10 dataset, they obtain a maximum F_1-score of 0.85 with almost equal recall and precision using the Logistic Regression Boosting classifier. When compared to a system based on meta data, reputation and lexical features solely, the shallow syntactic and semantic features introduced an improvement of around 3%.

Adler et al. [1] present a combination of previous approaches, resulting in a system based on meta data, textual and natural language features. Table 5.4 lists several of the features they use. Together with new results based on a meta-classifier which combines previously applied features, they provide an overview of existing work on vandalism detection and an extensive list of classification features. Furthermore, they distinguish between immediate and historic vandalism. The latter refers to vandalized revisions in an article's revision history. Some of their features can only be applied to historic vandalism, as they refer to subsequent revisions. The experiments are performed on the PAN-WVC 10 corpus. A ten-fold cross-validation with

a Random Forest classifier shows that the usage of features for historic vandalism increases performance from 0.969 to 0.976 AUC (area under the ROC curve). This is due to features like *next-comment-indicates-revert* and *time-until-next-edit*. As a possible reason for the efficiency of the latter feature, the authors state that frequently edited pages are more likely to be vandalized, without explicitly giving a source for that assumption. They conclude that the gain from using language features should not be overestimated, based on the fact that the calculation of language-specific features is generally more time-consuming than the calculation of textual, meta data or reputation features.

Javanmardi et al. [12] classify their features in a similar way, i.e. into meta data, textual and language model features and present their detailed descriptions. The last category is based on the Kullback-Leibler distance between two unigram language models. As in [1], they use the PAN-WVC 10 corpus in their experiments and divide it into training and test data. A Random Forest classifier performed best, yielding a maximum AUC value of 0.955 on the test data, and 0.974 with a 3-fold cross-validation on the training set. Javanmardi et al. experimented with the same corpus as [1]. However, their results are not comparable, as [1] employed cross-validation and Javanmardi et al. did not. The evaluation of different groups of features shows that their language model and meta data features are less significant than textual and reputation features. This partly contradicts the findings of Adler et al. [1], who find reputation features to be less important than other meta data features such as time or comment length. Javanmardi et al. explain this with differences in their definition of features. Furthermore, the classification approaches are different: Adler et al. use a meta-classifier combining different classifiers that have been developed for certain features, whereas Javanmardi et al. use one classifier trained and tested with different groups of features. To identify redundant individual features, the authors use a Logistic Regression approach. Among the most important individual features, they identify *Insert-Special-Words* (insertions of vulgarism, spam, sex etc. words) and a feature related to the author reputation.

We conclude that absolute agreement on the importance of different features for vandalism detection does not exist. In any case, vandalism detection depends on methods and algorithms developed to calculate the difference between adjacent revisions. Textual and language features use this kind of information, and revision meta data information has also proved to be helpful. Author reputation is in some cases bound to an edit history for authors, which also demands for a precalculation based on the revision histories. As already pointed out earlier in our survey, structured and systematic access to the type of edits performed in each revision, could also help the generation of features for vandalism detection. We consider this as a reference for future research.

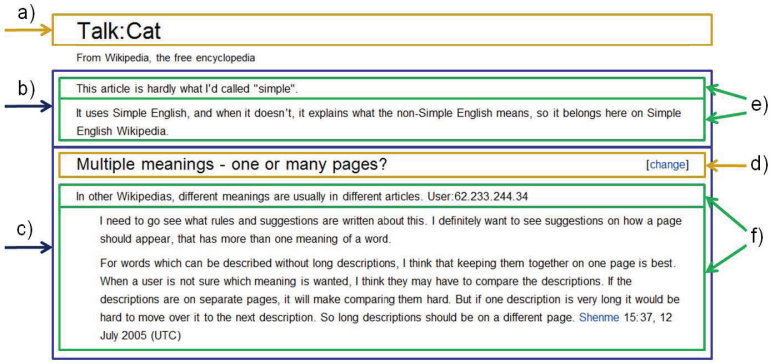


Fig. 5.4 Structure of a Talk page: *a)* Talk page title, *b)* untitled discussion topic, *c)* titled discussion topic, *d)* topic title, *e)* unsigned turns, *f)* signed turns

5.3 Discussions in Wikipedia

So far, we have shown that the revision history of Wikipedia is a valuable resource for many NLP applications. By regarding the whole evolution of an article rather than just its latest version, it is possible to leverage the dynamic properties of Wikipedia. The article revision history reflects a product-centered view of the collaborative writing process. In order to fully understand collaborative writing and, in turn, collaboratively constructed resources, is is necessary to include the writing process itself into the equation.

In joint writing, authors have to externalize processes that are not otherwise made explicit, like the planning and the organization of the text. Traditionally, these processes could only be observed indirectly by conducting interviews with the authors. In Wikipedia, however, coordination and planning efforts can be observed on the article Talk pages on which the Wikipedians discuss the further development of the articles (see Fig. 5.4). These discussion spaces are a unique resource for analyzing the processes involved in collaborative writing.

Technically speaking, a Talk page is a normal wiki page located in one of the Talk namespaces (see Table 5.5). Similar to a web forum, they are divided into discussions (or topics) and contributions (or turns). What distinguishes wiki discussions from a regular web forum, however, is the lack of a fixed, rigid thread structure. There are no dedicated formatting devices for structuring the Talk pages besides the regular wiki markup.

Each Talk page is implicitly connected to an article by its page name—e.g., the Talk page `Talk:Germany` corresponds to the article `Germany`. It is, however, not possible to establish explicit connections between individual discussions on the page and the section of the article that is being discussed. Each namespace in Wikipedia has a corresponding Talk namespace resulting in a total of ten different types of Talk pages (Table 5.5) which can be categorized into four functional classes:

Table 5.5 Wikipedia namespaces and functional Talk page classes

Basic namespaces	Talk namespaces	Functional class
Main	Talk	Article
User	User talk	User
Wikipedia	Wikipedia talk	Meta
MediaWiki	MediaWiki talk	Meta
Help	Help talk	Meta
File	File talk	Item
Template	Template talk	Item
Category	Category talk	Item
Portal	Portal talk	Item
Book	Book talk	Item

- **Article Talk pages** are mainly used for the coordination and planning of articles.
- **User Talk pages** are used as the main communication channel and social networking platform for the Wikipedians.
- **Meta Talk pages** serve as a platform for policy making and technical support.
- **Item-specific Talk pages** are dedicated to the discussion of individual media items (e.g., pictures) or structural devices (e.g., categories and templates).

The users are asked to structure their contributions using paragraphs and indentation. One *turn* may consist of one or more paragraphs, but no paragraph may span over several turns. Turns that reply to another contribution are supposed to be indented to simulate a thread structure. We call this *soft threading* as opposed to *explicit threading* in web forums.

Users are furthermore encouraged to append signatures to their contributions to indicate the end of a turn (see Fig. 5.5). There are extensive policies[20] that govern the usage and format of signatures. They usually should contain the username of the author and the time and date of the contribution. However, users' signatures do not adhere to a uniform format, which makes reliable parsing of user signatures a complex task. Moreover, less than 70% of all users explicitly sign their posts [35]. In some cases, depending on the setup of an individual Talk page, automatic scripts—so-called "bots"—take over whenever an unsigned comment is posted to a Talk page and add the missing signature. While this is helpful for signature-based discourse segmentation, it is misleading when it comes to author identification (see Fig. 5.5, signature 5.6).

Due to the lack of discussion-specific markup, contribution boundaries are not always clear-cut. They may even change over time, for instance if users insert their own comments into existing contributions of other users, which results in non-linear discussions (see Fig. 5.6). This makes automatic processing of Talk pages a challenging task and demands a substantial amount of preprocessing.

[20] `http://en.wikipedia.org/wiki/WP:SIGNATURE`

The Rambling Man (talk) 18:20, 27 February 2012 (UTC) (5.1)

–66.53.136.85 21:41, 2004 Aug 3 (UTC) (5.2)

– Taku (5.3)

– Preceding unsigned comment added by 121.54.2.122 (talk) 05:33, 10 February 2012 (UTC) (5.4)

–SineBot (talk) 08:43, 31 August 2009 (UTC) (5.5)

Imzadi 1979 > 09:20, 20 May 2011 (UTC) (5.6)

– ♫Greatorangepumpkin♫ T 14:14, 17 December 2010 (UTC) (5.7)

Fig. 5.5 Examples for user signatures on Talk pages: (5.1) Standard signature with username, link to user Talk page and timestamp (5.2) Signature of an anonymous user (5.3) Simple signature without timestamp (5.4,5.5) Bot-generated signatures (5.6,5.7) Signatures using colors and special unicode characters as design elements

There are ongoing attempts to improve the usability of the discussion spaces with extensions for explicit threading[21] and visual editing[22]. However, these enhancements have been tested in only selected small Wikimedia projects and have not yet been deployed to the larger wikis.

In order to prevent individual Talk pages from becoming too long and disorganized, individual discussions can be moved to a discussion archive[23]. Discussion archives are marked with an "Archive" suffix and usually numbered consecutively. The oldest discussion archive page for the article "Germany", for example, is named `Talk:Germany/Archive_1`. There are two possible procedures for archiving a Talk page: the *cut-and-paste procedure* and the *move procedure*. While it is not possible to determine directly which method has been used to create an archive, the choice has important implications for page processing. The cut-and-paste procedure copies the text from an existing Talk page to a newly created archive page.

Fig. 5.6 Inserted comments within user turn

[21] `http://www.mediawiki.org/wiki/Extension:LiquidThreads`

[22] `http://www.mediawiki.org/wiki/Visual_editor`

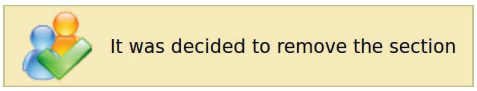
[23] `http://en.wikipedia.org/wiki/WP:ARCHIVE`

All revisions of this Talk page remain in the revision history of the original page. The move procedure renames (i.e., moves) an existing Talk page and adds the numbered archive suffix to its page title. Afterwards, a new Talk page is created that is then used as the new active Talk space. Archives created with the latter procedure maintain their own revision history, which simplifies the revision-based processing of these pages.

Even though there is no discussion-specific markup to structure Talk pages, so-called *templates* can be used to better organize the discussions. In their simplest form, templates are small wiki pages that can be embedded in another page using a shortcut. These templates are commonly used to embed info banners and predefined messages into the lead section of articles and Talk pages or to mark important decisions in a discussion. For example, the template

{{*consensus*|*It was decided to remove the section*}}

is replaced with



to highlight the consensus of a discussion. Depending on the individual template, the embedded content is either transcluded (i.e., inserted into the page on runtime but not in the source code), or substituted (i.e., inserted directly in the source code). While the latter approach is easier to process, most templates follow the transclusion method.

A specific subset of templates is used as a tagset for labeling articles and Talk pages. By adding the template {{*controversial*}} to a Talk page, an information banner is placed in the lead section of the Talk page and the associated article is tagged as controversial. A complete overview of Talk space specific templates can be found on the corresponding Wikipedia policy pages[24]. The cleanup and flaw markers are especially helpful criteria for filtering articles and Talk pages for corpus creation or further analysis.

The remainder of this section will discuss the merits of Talk pages as a resource for NLP and present a selection of qualitative and quantitative studies of discussions in Wikipedia. An overview can be found in Table 5.6.

5.3.1 Work Coordination and Conflict Resolution

Viégas et al. [35] were among the first to draw attention to Wikipedia Talk pages as an important resource. In an empirical study, they discovered that articles with Talk pages have, on average, 5.8 times more edits and 4.8 times more participating users than articles without any Talk activity. Furthermore, they found that the number of new Talk pages increased faster than the number of content pages. In order

[24] `http://en.wikipedia.org/wiki/WP:TTALK`

Table 5.6 Qualitative and quantitative analyses of Wikipedia Talk pages

Reference	Focus	Corpus Size	Wikipedia	Tagset
[35]	Coordination	25 TP	English	11
[29, 30]	Coordination	100 TP	English	15
[13]	Coordination, Conflict	–	English	–
[9]	Coordination, Information Quality	100 TP	Simple English	17
[32]	Information Quality	60 TP	English	12
[24]	Authority Claims	30 D	English	6
[2]	Authority Claims, Alignment Moves	47 TP	English	6[a], 8[b]
[15]	User Interaction	–	English	–
[17]	User Interaction	–	Venetian	–

[a] Authority Claims
[b] Alignment Moves (3 positive, 5 negative)
TP = Talk Pages
D = Discussions

to better understand how the rapidly increasing number of Talk pages are used by Wikipedians, they performed a qualitative analysis of selected discussions. The authors manually annotated 25 "purposefully chosen"[25] Talk pages with a set of 11 labels in order to analyze the aim and purpose of each user contribution. Each turn was tagged with one of the following labels:

- *request for editing coordination*
- *request for information*
- *reference to vandalism*
- *reference to Wikipedia guidelines*
- *reference to internal Wikipedia resources*
- *off-topic remark*
- *poll*
- *request for peer review*
- *information boxes*
- *images*
- *other*

The first two categories, requests for coordination (58.8%) and information (10.2%), were most frequently found in the the analyzed discussions, followed by off-topic remarks (8.5%), guideline references (7.9%), and references to internal resources (5.4%). This shows that Talk pages are not used just for the "retroactive resolution of disputes", as the authors hypothesized in their preliminary work [34]; rather, they are used for proactive coordination and planning of the editorial work.

Schneider et al. [29, 30] pick up on the findings of Viégas et al. and manually analyze 100 Talk pages with an extended annotation schema. In order to obtain a representative sample for their study, they define five article categories to choose the Talk pages from: *most-edited articles*, *most-viewed articles*, *controversial articles*,

[25] According to [35], "[t]he sample was chosen to include a variety of controversial and non-controversial topics and span a spectrum from hard science to pop culture."

featured articles, and a *random set of articles*. In addition to the 11 labels established in [35], Schneider et al. classify the user contributions as

- *references to sources outside Wikipedia*
- *references to reverts, removed material or controversial edits*
- *references to edits the discussant made*
- *requests for help with another article*

The authors evaluated the annotations from each category separately and found that the most frequent labels differ between the five classes. Characteristic peaks in the class distribution could be found for the "reverts" label, which is a strong indicator for discussions of controversial articles. Interestingly, the controversial articles did not have an above-average discussion activity, which was initially expected due to a high demand of coordination. The labels "off-topic", "info-boxes", and "info-requests" peak in the random category, which are apt to contain shorter Talk pages than the average items from the other classes. In accordance with [35], coordination requests are the most frequent labels in all article categories, running in the 50% to 70% range. The observed distribution patterns alone are not discriminative enough for identifying the type of article a Talk page belongs to, but they nevertheless serve as valuable features for Talk page analysis.

Furthermore, the labels can be used to filter or highlight specific contributions in a long Talk page to improve the usability of the Talk platform. In [30], the authors perform a user study in which they evaluate a system that allows discussants to manually tag their contribution with one of the labels. Most of the 11 participants in the study perceived this as a significant improvement in the usability of the Talk page, which they initially regarded as confusing. Given enough training data, this classification tasks can be tackled automatically using machine learning algorithms.

In a large-scale quantitative analysis, Kittur et al. [13] confirm earlier findings by [35] and demonstrate that the amount of work on content pages in Wikipedia is decreasing while the *indirect work* is on the rise. They define indirect work as "excess work in the system that does not directly lead to new article content." Besides the efforts for work coordination, indirect work comprises the resolution of conflicts in the growing community of Wikipedians. In order to automatically identify conflict hot spots or even to prevent future disputes, the authors developed a model of conflict on the article level and demonstrate that a machine learning algorithm can predict the amount of conflict in an article with high accuracy. In contrast to the works discussed above, Kittur et al. do not employ a hand-crafted coding schema to generate a manually annotated corpus; rather, they extract the *"controversial"* tags that have been assigned to articles with disputed content by Wikipedia editors. This human-labeled conflict data is obtained from a full Wikipedia dump with all page revisions (*revision dump*) using the Hadoop[26] framework for distributed processing. The authors define a measure called *Controversial Revision Count* (CRC) as "the number of revisions in which the 'controversial' tag was applied to the article". These scores are used as a proxy for the amount of conflict in a specific article and

[26] `http://hadoop.apache.org/`

Table 5.7 Page-level features proposed in [13]

Feature	Page
Revisions[a]	Article[4], Talk[1], Article/Talk
Page length	Article, Talk, Article/Talk
Unique editors[a]	Article[5], Talk, Article/Talk
Unique editors[a]/Revisions[a]	Article, Talk[3]
Links from other articles[a]	Article, Talk
Links to other articles[a]	Article, Talk
Anonymous edits[a,b]	Article[7], Talk[6]
Administrator edits[a,b]	Article, Talk
Minor edits[a,b]	Article, Talk[2]
Reverts[a,c]	Article

[a] Raw counts
[b] Percentage
[c] By unique editors
[1–7] Feature utility rank

are predicted by a Support Vector Machine regression algorithm from raw data. The model is trained on all articles that are marked as controversial in their latest revision and evaluated by means of five-fold cross validation. As features, the authors define a set of page- level metrics based on both articles and talk pages (see Table 5.7). They evaluated the usefulness of each feature, which is indicated by the individual ranks as superscript numbers in the table.

The authors report that the model was able to account for almost 90% of the variation in the CRC scores ($R^2 = 0.897$). They furthermore validate their model in a user study by having Wikipedia administrators evaluate the classification results on 28 manually selected articles that have not been tagged as controversial. The results of this study showed that the CRC model generalizes well to articles that have never been tagged as controversial. This opens up future applications like identifying controversial articles before a critical point is reached.

5.3.2 Information Quality

The information quality (IQ) of collaboratively constructed knowledge resources is one of their most controversially disputed aspects. These resources break with the traditional paradigm of editorial quality assurance usually found in expert-mediated knowledge bases and allow anyone to view and edit the information at their discretion. As collaboratively constructed resources become increasingly important, it is a vital necessity to measure their quality to ensure the reliability and thus the trustworthiness of their content. In section 5.2.3, we already discussed how the revision history has been used to assess the article quality and trustworthiness using Wikipedia internal quality categories as frames of references. The assumption is that any article similar to a known high quality article with respect to the features defined by the

individual assessment approach is again of high quality. While this approach is useful for evaluating different types features as to how they are able to quantify quality related aspects, the notion of quality itself is somewhat limited, since it provides no insight into the concrete problems a given article suffers from. In order to assess the information quality and potential quality problems of an article, a more fine grained concept of quality is needed. In Wikipedia, the information in Talk pages contains valuable insights into the readers' judgments of articles and comments about their potential deficiencies. Consequently, an analysis of these Talk pages with respect to article information quality is a good starting point for establishing a fine grained quality assessment model for Wikipedia.

From an information scientific perspective, Stvilia et al. [32] raise the question of how quality issues in Wikipedia are discussed by the community and how the open and unstructured discussions on Talk pages can be an integral part of a successful quality assurance process. The authors manually analyze 60 discussion pages in order to identify which types of IQ problems have been discussed by the community. They determine twelve IQ problems along with a set of related causal factors for each problem and actions that have been suggested by the community to tackle them.

For instance, IQ problems in the quality dimension *complexity* may be caused by low readability or complex language and might be tackled by replacing, rewriting, simplifying, moving, or summarizing the problematic article content. They furthermore identify trade-offs among these quality dimensions of which the discussants on Talk pages are largely aware. For example, an improvement in the dimension *completeness* might result in a deterioration in the *complexity* dimension. This model of IQ problems is useful for NLP applications in two ways: (1) as a frame of reference for automatic quality assessment of collaboratively created content, and (2) for automatically improving its quality using NLP techniques. In order to measure quality automatically, it is important to define what quality is and how it can be measured. Here, the proposed model is a sound basis for grounding any language processing approach to quality assessment with the users' understanding of quality. In order to use automatically calculated scores to improve article quality automatically, it is necessary to identify which actions can be taken to increase the quality scores in each dimension. This is also provided by the model proposed in [32].

Ferschke et al. [9] take the next step towards an automatic analysis of the discussions in Wikipedia. Inspired by the aforementioned IQ model, they develop an annotation schema for the discourse analysis of Talk pages aimed at the coordination effort for article improvement (see Table 5.8). With 17 labels in four categories, the schema captures article criticism and explicit user actions aimed at resolving IQ problems as well as the flow of information and the attitude of the discussants towards each other. The authors create a corpus of 100 Talk pages from the Simple English Wikipedia which they automatically segmented into individual discussions and turns by using the revision history for identifying turn boundaries and for attributing the correct authors without relying on user signatures. They manually label the corpus using their annotation schema and report a chance-corrected inter-annotator agreement between two raters of $\kappa = 0.67$ over all labels. In order to automatically

Table 5.8 Annotation schema for the discourse analysis of Wikipedia Talk pages proposed in [9]

Article criticism	Explicit performative	Information content	Interpersonal
Missing content	Suggestion	Information providing	Positive (+)
Incorrect content	Reference to resource	Information seeking	Partially +/−
Unsuitable content	Commitment to action	Information correcting	Negative (−)
Structural problem	Report of action		
Stylistic problem			
Objectivity issues			
Other			

label the turns in unseen Talk pages, the authors use the annotated corpus as training data for a set of machine learning algorithms and train individual classifiers for each label. They combine the best performing classification models into a classification pipeline which they use to label untagged discussions. They report an overall classification performance of $F_1 = 0.82$ evaluated on ten-fold cross-validation. The automatic classification of turns in Wikipedia Talk pages is a necessary prerequisite to investigating the relations between article discussions and article edits, which, in turn, is an important step towards understanding the processes of collaboration in large-scale wikis. Moreover, it enables practical applications that help to bring the content of Talk pages to the attention of article readers.

5.3.3 Authority and Social Alignment

Information quality discussions in Wikipedia can have a big impact on articles. They usually aim at keeping articles in line with Wikipedia's guidelines for quality, neutrality and notability. If such a discussion is not grounded on authoritative facts but rather on subjective opinions of individual users, a dispute about content removal, for example, may lead to the unjustified removal of valuable information. Wikipedia Talk pages are, for the most part, pseudonymous discussion spaces and most of the discussants do not know each other personally. This raises the question how the users of Talk pages decide which claim or statement in a discussion can be trusted and whether an interlocutor is reliable and qualified.

Oxley et al. [24] analyze how users establish credibility on Talk pages. They define six categories of *authority claims* with which users account for their reliability and trustfulness (see Table 5.9). Based on this classification, Bender et al. [2] created a corpus of social acts in Wikipedia Talk pages (AAWD). In addition to authority claims, the authors define a second annotation layer to capture *alignment moves*—i.e., expressions of solidarity or signs of disagreement among the discussants. At least two annotators labeled each of the 5,636 turns extracted from 47 randomly sampled Talk pages from the English Wikipedia. The authors report an overall inter-annotator agreement of $\kappa = 0.59$ for authority claims and $\kappa = 0.50$ for alignment moves.

Table 5.9 Authority claims proposed in [24, 2]

Claim type	Based on
Credentials	Education, Work experience
Experiential	Personal involvement in an event
Institutional[a]	Position within the organizational structure
Forum	Policies, Norms, Rules of behavior (in Wikipedia)
External	Outside authority or resource
Social Expectations	Beliefs, Intentions, Expectations of social groups (outside of Wikipedia)

[a] Not encoded in the AAWD corpus

Marin et al. [16] use the AAWD corpus to perform machine learning experiments targeted at automatically detecting authority claims of the *forum* type (cf. Table 5.9) in unseen discussions. They particularly focus on exploring strategies for extracting lexical features from sparse data. Instead of relying only on *n*-gram features, which are prone to overfitting when used with sparse data, they employ knowledge-assisted methods to extract meaningful lexical features. They extract word lists from Wikipedia policy pages to capture policy-related vocabulary and from the articles associated with the Talk pages to capture vocabulary related to editor discussions. Furthermore, they manually create six word lists related to the labels in the annotation schema. Finally, they augment their features with syntactic context gained from parse trees in order to incorporate a higher level linguistic context and to avoid the explosion of the lexical feature space that is often a side effect of higher level *n*-grams. Based on these features, the authors train a maximum entropy classifier to decide for each sentence whether it contains a forum claim or not.[27] The decision is then propagated to the turn level if the turn contains at least one forum claim. The authors report an F_1-score for the evaluation set of 0.66.

Besides being a potential resource for social studies and online communication research, the AAWD corpus and approaches to automatic classification of social acts can be used to identify controversial discussions and online trolls.[28]

5.3.4 User Interaction

It is not only the content of Talk pages which has been the focus of recent research, but also the social network of the users who participate in the discussions. Laniado et al. [15] create Wikipedia discussion networks from Talk pages in order to capture structural patterns of interaction. They extract the thread structure from all article and user Talk pages in the English Wikipedia and create tree structures of the discussions. For this, they rely on user signatures and turn indentation. The au-

[27] The corpus was split into training set (67%), development set (17%) and test set (16%).

[28] A *troll* is a participant in online discussions with the primary goal of posting disruptive, off-topic messages or provoking emotional responses.

thors consider only registered users, since IP addresses are not unique identifiers for the discussants. In the directed article reply graph, a user node A is connected to a node B if A has ever written a reply to any contribution from B on any article Talk page. They furthermore create two graphs based on User Talk pages which cover the interactions in the personal discussion spaces in a similar manner.

The authors analyze the directed degree assortativity of the extracted graphs. In the article discussion network, they found that users who reply to many different users tend to interact mostly with inexperienced Wikipedians while users who receive messages from many users tend to interact mainly with each other. They furthermore analyzed the discussion trees for each individual article, which revealed characteristic patterns for individual semantic fields. This suggests that tree representations of discussions are a good basis for metrics for characterizing different types of Talk pages while the analysis of User Talk pages might be a good foundation for identifying social roles by comparing the different discussion fingerprints of the users.

A different aspect of the social network analysis in Wikipedia is examined by Massa [17]. He aims at reliably extracting social networks from User Talk pages. Similarly to [15], he creates a directed graph of user interactions. The interaction strength between two users is furthermore quantified by weighted edges with weights derived from the number of messages exchanged by the users. The study is based on networks extracted from the Venetian Wikipedia. Massa employs two approaches to extract the graphs automatically, one based on parsing user signatures and the other based on the revision history. He compares the results with a manually created gold standard and found that the revision based approach produces more reliable results than the signature approach, which suffers from the extreme variability of the signatures. However, history based processing often resulted in higher weights of the edges, because several edits of a contribution are counted as individual messages. A history-based algorithm similar to the one used by Ferschke et. al [9] could account for this problem. Massa furthermore identifies several factors that impede the network extraction, like noise in form of bot messages and vandalism, inconsistently used usernames, and unsigned messages. While these insights might be a good basis for future work on network extraction tasks, they are limited by the small Venetian Wikipedia on which the study is based. Talk pages in larger Wikipedias are much longer, more complex and are apt to contain pitfalls not recognized by this work.

5.4 Tools and Resources

In the following, we describe tools for processing revisions and discussions from Wikipedia as well as corpora which offer this content in a structured form.

Table 5.10 Tools for accessing Wikipedia articles, revisions and Talk pages

Reference and Name	Type of Data	API	License
MediaWiki API	pages and revisions	web service	–
JWPL [44]	pages (incl. Talk)	Java	LGPL
Wikipedia Revision Toolkit [8]	revisions	Java	LGPL
Wikipedia Miner [20]	articles	Java	GPL
WikiXRay [31]	quantitative statistics	Python, R	GPL

5.4.1 Tools for Accessing Wikipedia Articles, Revisions and Talk Pages

We give an overview of tools for accessing and processing Wikipedia articles, revisions, discussions or statistics about them. All of them are freely available, some are open-source. This list does *not* include special purpose scripts[29] provided by Wikipedia users or individual projects hosted on the Wikimedia Toolserver (see below).

The MediaWiki API[30] provides direct access to MediaWiki databases including Wikipedia. It can be accessed via a web service[31] or various client code wrappers[32]. Many bots and Toolserver utilities use this facility to get the data they need and to edit pages. The MediaWiki API supports various actions like *query*, *block* or *edit* and output formats such as JSON, PHP or XML. As it works directly on the MediaWiki databases, the API provides real time access to Wikipedia. This is a discriminative feature comparing it to any other API that is working on static dumps (cf. section 5.4.2).

The Java Wikipedia Library (JWPL) [44] offers a Java-based programming interface for accessing all information in different language versions of Wikipedia in a structured manner. It includes a MediaWiki markup parser for in-depth analysis of page contents. JWPL works with a database in the background, the content of the database comes from a dump, i.e. a static snapshot of a Wikipedia version. JWPL offers methods to access and process properties like in- and outlinks, templates, categories, page text —parsed and plain— and other features. The *Data Machine* is responsible for generating the JWPL database from raw dumps. Depending on what data are needed, different dumps can be used, either including or excluding the Talk page namespace.

The Wikipedia Revision Toolkit [8] expands JWPL with the ability to access Wikipedia's revision history. To this end, it is divided into two tools, the *TimeMachine* and the *RevisionMachine*. The TimeMachine is capable of restoring any past state of the encyclopedia, including a user-defined interval of past versions of the

[29] A compilation of these can be found under `http://en.wikipedia.org/wiki/WP:WikiProject_User_scripts/Scripts`

[30] `http://www.mediawiki.org/wiki/API`

[31] `http://en.wikipedia.org/w/api.php`

[32] `http://www.mediawiki.org/wiki/API:Client_code`

pages. The RevisionMachine provides access to the entire revision history of all Wikipedia articles. It stores revisions in a compressed form, keeping only differences between adjacent revisions. The Revision Toolkit additionally provides an API for accessing Wikipedia revisions along with meta data like the comment, timestamp and information about the user who made the revision.

Wikipedia Miner [20] offers a Java-based toolkit to access and process different types of information contained in Wikipedia articles. Similar to JWPL, it has an API for structured access to basic information of an article. Categories, links, redirects and the article text, plain or as MediaWiki markup, can also be accessed as Java classes. It runs a preprocessed Java Berkeley database in the background to store the information contained in Wikipedia. Wikipedia Miner has a focus on concepts and semantic relations within Wikipedia. It is able to detect and sense-disambiguate Wikipedia topics in documents, i.e. it can be used to wikify plain text. Furthermore, the framework compares terms and concepts in Wikipedia, calculating their semantic relatedness or related concepts based on structural article properties (e.g. in-links) or machine learning. In contrast to JWPL, it cannot be used to access and process the revision history of an article. The capability of its parser is rather limited, e.g. no templates or infoboxes can be processed.

WikiXRay [31] is a collection of Python and GNU R scripts for the quantitative analysis of Wikipedia data. It parses plain Wikimedia dumps and imports the extracted data into a database. This database is used to provide general quantitative statistics about editors, pages and revisions.

Finally, the Wikimedia Toolserver[33] is a hosting platform for tools dedicated to processing Wikimedia data. The tools and scripts on the Toolserver are mainly developed by Wikipedia editors and researchers for Wiki maintenance and analysis. The unique advantage of running software on the Toolserver is the direct access to data from mirrored Wikimedia databases. The databases offer more information than the downloadable data dumps and are always kept up-to-date. However, computing resources are limited, so that the Toolserver is not an appropriate platform for running applications that demand much processing power.

5.4.2 Resources based on Data from Wikipedia's Article and Talk Pages

This paragraph assembles a list of corpora containing either data directly exported from Wikipedia pages, revisions or discussion pages, or data that has been extracted and annotated from one of these sources for different tasks. Rather than being exhaustive, this list is meant to give a short overview of existing data collections that have been introduced in the course of this chapter and are freely available. The resources we presented can roughly be divided into corpora produced from the article revisions and from Talk pages. Table 5.11 provides an overview.

[33] `http://toolserver.org/`

Table 5.11 Resources based on Wikipedia articles and Talk pages

Resource	Based on	Annotations	Format	License
WVC [26, 27]	revisions	vandalism	CSV	CC
WiCoPaCo [18]	revisions	spelling errors and paraphrases	XML	GFDL
[40]	revisions	lexical simplifications	CSV	–
[41]	revisions	textual entailment	XML, TXT	–
[43]	revisions	real-word spelling errors	TXT	CC
SEWD Corpus [9]	discussions	dialog acts	XMI, MMAX	CC
AAWD Corpus [2]	discussions	social acts	XTDF	–

The main source of raw data from Wikipedia is usually one of the so called Wikimedia *dumps*[34]. These dumps are snapshots of different content from Wikimedia Wiki projects, usually stored in large compressed XML and SQL files, with various releases throughout a year. The XML dumps store text including MediaWiki markup and metadata, separated by namespace. The main *page data* dumps are usually divided into three sets: *pages-articles* contains current versions of pages excluding Talk- and User-pages, *pages-meta-current* contains current page versions including Talk- and user-pages and *pages-meta-history* contains all revisions of all pages. Besides the tools mentioned in section 5.4.1, there are various programs[35] available to handle Wikipedia dumps, all of them being limited to downloading or importing the dumps into databases for further processing.

5.5 Conclusion

For the last several years, the importance of Wikipedia in academic research has been continuously growing. In particular, NLP researchers increasingly find it to be a valuable resource to analyze the process of collaborative writing. To demonstrate this, we focused on the dynamic aspects of Wikipedia—that is, on the fact that its content is constantly changing. The massive amount of data that is generated by storing each edit to any page in Wikipedia offers numerous possibilities to create task-specific corpora, such as training data for tasks such as spelling error detection and information quality assessment. Although the number of studies on this kind of data has increased, to the best of our knowledge, there is no comprehensive introduction to existing applications in this field. In this survey, we therefore sought to analyze and compare methods for analyzing the process of collaborative writing based on Wikipedia's revision history and its discussion pages.

Section 5.2 described the concept of page revisions in Wikipedia. After defining the necessary terms, we explained various approaches generating training data for NLP tasks from the semistructured revision history data. Most of these approaches

[34] `http://dumps.wikimedia.org/`

[35] `http://meta.wikimedia.org/wiki/Data_dumps#Tools`

are either related to spelling error detection and correction or paraphrasing. A common way to process revision data is to calculate the changes (i.e., edits) between adjacent revisions and subsequently select suitable edit examples for further processing. The latter can be done by applying filters either on the raw edit data or after post-processing the contained lexical or syntactical information. Furthermore, approaches differ in whether they keep or ignore wiki markup such as links and headlines. Another series of approaches using revision history data aims to assess article quality. We distinguished different studies by the type of revision features they employ and by their definition of article quality. Revision features include quantitative properties like raw edit counts, but also various concepts of stability of article contents. Article quality criteria are mostly based on the Wikipedia internal review system. While article quality is an important factor for the reliability of the encyclopedic content in Wikipedia, vandalism is a serious problem to address. Vandalism detection is the task of distinguishing between valid and malicious edits in Wikipedia. It thus naturally uses revision history data, as the decision whether an edit is vandalistic or not is most likely based on the analysis of the changes between one revision and another. Advanced vandalism detection algorithms are based on machine learning and thus utilize a wide range of features. Typical vandalism features are based on changes in the article and/or on meta data. We compared both types of approaches.

Collaboration in a large text collection like Wikipedia is a demanding task and therefore needs coordination. Wikipedia offers a space for discussion among the authors, the so-called Talk pages. Whereas information from the revision history in Wikipedia is mainly used to support specific NLP applications (e.g. by augmenting the amount of training data), Wikipedia discussions are mostly analyzed to find out more about the process of collaboration in Wikipedia. Section 5.3 introduced the concept of Talk pages in Wikipedia and explained the challenges related to their processing. We analyzed various types of quantitative and qualitative NLP studies of Wikipedia discussions. A number of them focus on the utility of Talk pages for coordination and conflict resolution among the authors of an article. We introduced approaches using labels to categorize the purpose of contributions in discussions. They agree in the finding that coordination requests are the most frequent type of contributions. Another approach uses a machine learning model which is, amongst others, based on Talk page features to identify highly controversial articles. As an extension to the work discussed in Sect. 5.2.3, we reported on approaches analyzing the information quality of Wikipedia contents based on discussion page properties. We presented two studies with a focus on quality assessment based on a qualitative analysis of Talk page contributions. Both of them developed a model of information quality useful to NLP applications. We then turned to the social aspects of the discussion pages in Wikipedia. First, we introduced a corpus of so called social acts in Talk pages, along with various studies based on authority and social alignment among Wikipedia authors. Second, we explained two approaches investigating a network of user interaction in Wikipedia based on the thread structure of Talk pages.

A summary of tools and corpora for accessing and processing collaboratively constructed discourse in Wikipedia is presented in Sect. 5.4. We explained different

methods and tools to access Wikipedia revisions and/or Talk pages. Additionally, we gave a summary of the freely accessible corpora presented in Sects. 5.2 and 5.3.

We have discussed approaches exploiting Wikipedia's revision history and its Talk pages. However, to understand the process of collaborative writing in Wikipedia even better, edit history information and discussion page contents should be brought together in future work. For example, it would be necessary to establish links between coordination and conflict resolution efforts on Talk pages and edits on the article. The resulting correlations could possibly answer a couple of very interesting questions with regard to the relevance or success of Wikipedia discussions. For example, it might be interesting to analyze the numbers of topics discussed on Talk pages which have actually been addressed by edits on the article itself. We think that this is a promising direction for future investigations based on the findings we presented in this survey.

Acknowledgments

This work has been supported by the Volkswagen Foundation as part of the Lichtenberg-Professorship Program under grant No. I/82806, and by the Hessian research excellence program "Landes-Offensive zur Entwicklung Wissenschaftlich-ökonomischer Exzellenz" (LOEWE) as part of the research center "Digital Humanities". We thank the anonymous reviewers for their valuable comments.

References

[1] Adler BT, Alfaro L, Mola-Velasco SM, Rosso P, West AG (2011) Wikipedia Vandalism Detection: Combining Natural Language, Metadata, and Reputation Features. In: Gelbukh A (ed) Computational Linguistics and Intelligent Text Processing, Springer, Lecture Notes in Computer Science, pp 277–288

[2] Bender EM, Morgan JT, Oxley M, Zachry M, Hutchinson B, Marin A, Zhang B, Ostendorf M (2011) Annotating Social Acts: Authority Claims and Alignment Moves in Wikipedia Talk Pages. In: Proceedings of the Workshop on Language in Social Media, Portland, OR, USA, pp 48–57

[3] Buriol LS, Castillo C, Donato D, Leonardi S, Millozzi S (2006) Temporal Analysis of the Wikigraph. In: Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence, Hong Kong, China, pp 45–51

[4] Chin SC, Street WN, Srinivasan P, Eichmann D (2010) Detecting Wikipedia Vandalism With Active Learning and Statistical Language Models. In: Proceedings of the 4th Workshop on Information Credibility, Hyderabad, India

[5] Cusinato A, Della Mea V, Di Salvatore F, Mizzaro S (2009) QuWi: Quality Control in Wikipedia. In: Proceedings of the 3rd Workshop on Information Credibility on the Web, ACM, Madrid, pp 27–34

[6] Dalip DH, Gonçalves MA, Cristo M, Calado P (2009) Automatic Quality Assessment of Content Created Collaboratively by Web Communities. In: Proceedings of the Joint International Conference on Digital Libraries, Austin, TX, USA, pp 295–304

[7] Emigh W, Herring SC (2005) Collaborative Authoring on the Web: A Genre Analysis of Online Encyclopedias. In: Proceedings of the 38th Annual Hawaii International Conference on System Sciences, Waikoloa, Big Island, HI, USA

[8] Ferschke O, Zesch T, Gurevych I (2011) Wikipedia Revision Toolkit: Efficiently Accessing Wikipedia's Edit History. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. System Demonstrations, Portland, OR

[9] Ferschke O, Gurevych I, Chebotar Y (2012) Behind the Article: Recognizing Dialog Acts in Wikipedia Talk Pages. In: Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics, Avignon, France

[10] Giampiccolo D, Trang Dang H, Magnini B, Dagan I, Cabrio E, Dolan B (2007) The Third PASCAL Recognizing Textual Entailment Challenge. In: Proceedings of the ACLPASCAL Workshop on Textual Entailment and Paraphrasing, Prague, Czech Republic, pp 1–9

[11] Han J, Wang C, Jiang D (2011) Probabilistic Quality Assessment Based on Article's Revision History. In: Proceedings of the 22nd International Conference on Database and Expert Systems Applications, Toulouse, France, pp 574–588

[12] Javanmardi S, McDonald DW, Lopes CV (2011) Vandalism Detection in Wikipedia: A High-Performing, Feature-Rich Model and its Reduction Through Lasso. In: Proceedings of the 7th International Symposium on Wikis and Open Collaboration, Mountain View, CA, USA, pp 82–90

[13] Kittur A, Suh B, Pendleton B, Chi EH (2007) He Says, She Says: Conflict and Coordination in Wikipedia. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, San Jose, CA, USA, pp 453–462

[14] Knight K, Marcu D (2000) Statistics-Based Summarization - Step One: Sentence Compression. In: Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence, Austin, TX, USA, pp 703–710

[15] Laniado D, Tasso R, Kaltenbrunner A, Milano P, Volkovich Y (2011) When the Wikipedians Talk : Network and Tree Structure of Wikipedia Discussion Pages. In: Proceedings of the 5th International Conference on Weblogs and Social Media, Barcelona, Spain, pp 177–184

[16] Marin A, Zhang B, Ostendorf M (2011) Detecting Forum Authority Claims in Online Discussions. In: Proceedings of the Workshop on Languages in Social Media, Portland, OR, USA, pp 39–47

[17] Massa P (2011) Social Networks of Wikipedia. In: Proceedings of the 22nd ACM Conference on Hypertext and Hypermedia, Eindhoven, Netherlands, pp 221–230

[18] Max A, Wisniewski G (2010) Mining Naturally-occurring Corrections and Paraphrases from Wikipedia's Revision History. In: Proceedings of the 7th

Conference on International Language Resources and Evaluation, Valletta, Malta
[19] Medelyan O, Milne D, Legg C, Witten IH (2009) Mining meaning from Wikipedia. International Journal of Human-Computer Studies 67(9):716–754
[20] Milne D, Witten IH (2009) An Open-source Toolkit for Mining Wikipedia. In: Proceedings of the New Zealand Computer Science Research Student Conference, Auckland, New Zealand
[21] Mizzaro S (2003) Quality control in scholarly publishing: A new proposal. Journal of the American Society for Information Science and Technology 54(11):989–1005
[22] Nelken R, Shieber SM (2006) Towards robust context-sensitive sentence alignment for monolingual corpora. In: Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics, Trento, Italy
[23] Nelken R, Yamangil E (2008) Mining Wikipedia's Article Revision History for Training Computational Linguistics Algorithms. In: Proceedings of the 1st AAAI Workshop on Wikipedia and Artificial Intelligence, Chicago, IL, USA
[24] Oxley M, Morgan JT, Hutchinson B (2010) "What I Know Is...": Establishing Credibility on Wikipedia Talk Pages. In: Proceedings of the 6th International Symposium on Wikis and Open Collaboration, Gdańsk, Poland, pp 2–3
[25] Posner IR, Baecker RM (1992) How People Write Together. In: Proceedings of the 25th Hawaii International Conference on System Sciences, Wailea, Maui, HI, USA, pp 127–138
[26] Potthast M (2010) Crowdsourcing a Wikipedia Vandalism Corpus. In: Proceedings of the 33rd Annual International ACM SIGIR Conference on Research & Development on Information Retrieval, Geneva
[27] Potthast M, Holfeld T (2011) Overview of the 2nd International Competition on Wikipedia Vandalism Detection. In: Notebook Papers of CLEF 2011 Labs and Workshops, Amsterdam, Netherlands
[28] Potthast M, Stein B, Gerling R (2008) Automatic Vandalism Detection in Wikipedia. In: Proceedings of the 30th European Conference on Advances in Information Retrieval, Glasgow, Scotland, UK, pp 663–668
[29] Schneider J, Passant A, Breslin JG (2010) A Content Analysis: How Wikipedia Talk Pages Are Used. In: Proceedings of the 2nd International Conference of Web Science, Raleigh, NC, USA, pp 1–7
[30] Schneider J, Passant A, Breslin JG (2011) Understanding and Improving Wikipedia Article Discussion Spaces. In: Proceedings of the 2011 ACM Symposium on Applied Computing, Taichung, Taiwan, pp 808–813
[31] Soto J (2009) Wikipedia: A Quantitative Analysis. PhD thesis
[32] Stvilia B, Twidale MB, Smith LC, Gasser L (2008) Information Quality Work Organization in Wikipedia. Journal of the American Society for Information Science and Technology 59(6):983–1001
[33] Thomas C, Sheth AP (2007) Semantic Convergence of Wikipedia Articles. In: Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence, Washington, DC, USA, pp 600–606

[34] Viégas FB, Wattenberg M, Dave K (2004) Studying Cooperation and Conflict Between Authors with History Flow Visualizations. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Vienna, Austria, pp 575–582

[35] Viégas FB, Wattenberg M, Kriss J, Ham F (2007) Talk Before You Type: Coordination in Wikipedia. In: Proceedings of the 40th Annual Hawaii International Conference on System Sciences, Big Island, HI, USA, pp 78–78

[36] Wang WY, McKeown KR (2010) Got you!: Automatic Vandalism Detection in Wikipedia with Web-based Shallow Syntactic-semantic Modeling. In: Proceedings of the 23rd International Conference on Computational Linguistics, Beijing, China, pp 1146–1154

[37] Wilkinson DM, Huberman BA (2007) Cooperation and Quality in Wikipedia. In: Proceedings of the 2007 International Symposium on Wikis, Montreal, Canada, pp 157–164

[38] Woodsend K, Lapata M (2011) Learning to Simplify Sentences with Quasi-Synchronous Grammar and Integer Programming. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Edinburgh, Scotland, UK, pp 409–420

[39] Yamangil E, Nelken R (2008) Mining Wikipedia Revision Histories for Improving Sentence Compression. In: Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. Short Papers, Association for Computational Linguistics, Columbus, OH, USA, pp 137–140

[40] Yatskar M, Pang B, Danescu-Niculescu-Mizil C, Lee L (2010) For the Sake of Simplicity: Unsupervised Extraction of Lexical Simplifications from Wikipedia. In: Proceedings of the 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Los Angeles, CA, USA, pp 365–368

[41] Zanzotto FM, Pennacchiotti M (2010) Expanding Textual Entailment Corpora from Wikipedia Using Co-Training. In: Proceedings of the 2nd COLING-Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources, Beijing, China

[42] Zeng H, Alhossaini MA, Ding L, Fikes R, McGuinness DL (2006) Computing Trust from Revision History. In: Proceedings of the 2006 International Conference on Privacy, Security and Trust, Markham, Ontario, Canada, pp 1–10

[43] Zesch T (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, Avignon, France

[44] Zesch T, Müller C, Gurevych I (2008) Extracting Lexical Semantic Knowledge from Wikipedia and Wiktionary. In: Proceedings of the 6th International Conference on Language Resources and Evaluation, Marrakech, Morocco

[45] Zhu Z, Bernhard D, Gurevych I (2010) A Monolingual Tree-based Translation Model for Sentence Simplification. In: Proceedings of the 23rd International Conference on Computational Linguistics, Beijing, China, pp 1353–1361

[46] Zobel J, Dart P (1996) Phonetic String Matching : Lessons from Information Retrieval. In: Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Zurich, Switzerland, pp 166–172