User Edits Classification Using Document Revision Histories

Amit Bronner
Informatics Institute
University of Amsterdam
a.bronner@uva.nl

Christof Monz
Informatics Institute
University of Amsterdam
c.monz@uva.nl

Abstract

Document revision histories are a useful and abundant source of data for natural language processing, but selecting relevant data for the task at hand is not trivial. In this paper we introduce a scalable approach for automatically distinguishing between factual and fluency edits in document revision histories. The approach is based on supervised machine learning using language model probabilities, string similarity measured over different representations of user edits, comparison of part-of-speech tags and named entities, and a set of adaptive features extracted from large amounts of unlabeled user edits. Applied to contiguous edit segments, our method achieves statistically significant improvements over a simple yet effective edit-distance baseline. It reaches high classification accuracy (88%) and is shown to generalize to additional sets of unseen data.

1 Introduction

Many online collaborative editing projects such as Wikipedia[1] keep track of complete revision histories. These contain valuable information about the evolution of documents in terms of content as well as language, style and form. Such data is publicly available in large volumes and constantly growing. According to Wikipedia statistics, in August 2011 the English Wikipedia contained 3.8 million articles with an average of 78.3 revisions per article. The average number of revision edits per month is about 4 million in English and almost 11 million in total for all languages.[2]

Exploiting document revision histories has proven useful for a variety of natural language processing (NLP) tasks, including sentence compression (Nelken and Yamangil, 2008; Yamangil and Nelken, 2008) and simplification (Yatskar et al., 2010; Woodsend and Lapata, 2011), information retrieval (Aji et al., 2010; Nunes et al., 2011), textual entailment recognition (Zanzotto and Pennacchiotti, 2010), and paraphrase extraction (Max and Wisniewski, 2010; Dutrey et al., 2011).

The ability to distinguish between factual changes or edits, which alter the meaning, and fluency edits, which improve the style or readability, is a crucial requirement for approaches exploiting revision histories. The need for an automated classification method has been identified (Nelken and Yamangil, 2008; Max and Wisniewski, 2010), but to the best of our knowledge has not been directly addressed. Previous approaches have either applied simple heuristics (Yatskar et al., 2010; Woodsend and Lapata, 2011) or manual annotations (Dutrey et al., 2011) to restrict the data to the type of edits relevant to the NLP task at hand. The work described in this paper shows that it is possible to automatically distinguish between factual and fluency edits. This is very desirable as it does not rely on heuristics, which often generalize poorly, and does not require manual annotation beyond a small collection of training data, thereby allowing for much larger data sets of revision histories to be used for NLP research.

In this paper, we make the following novel contributions:

We address the problem of automated classification of user edits as factual or fluency edits

[1] http://www.wikipedia.org

[2] Average for the 5 years period between August 2006 and August 2011. The count includes edits by registered users, anonymous users, software bots and reverts. Source: http://stats.wikimedia.org.

Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics, pages 356–366,
Avignon, France, April 23 - 27 2012.

by defining the scope of user edits, extracting a large collection of such user edits from the English Wikipedia, constructing a manually labeled dataset, and setting up a classification baseline.

A set of features is designed and integrated into a supervised machine learning framework. It is composed of language model probabilities and string similarity measured over different representations, including part-of-speech tags and named entities. Despite their relative simplicity, the features achieve high classification accuracy when applied to contiguous edit segments.

We go beyond labeled data and exploit large amounts of unlabeled data. First, we demonstrate that the trained classifier generalizes to thousands of examples identified by user comments as specific types of fluency edits. Furthermore, we introduce a new method for extracting features from an evolving set of unlabeled user edits. This method is successfully evaluated as an alternative or supplement to the initial supervised approach.

2 Related Work

The need for user edits classification is implicit in studies of Wikipedia edit histories. For example, Viegas et al. (2004) use revision size as a simplified measure for the change of content, and Kittur et al. (2007) use metadata features to predict user edit conflicts.

Classification becomes an explicit requirement when exploiting edit histories for NLP research. Yamangil and Nelken (2008) use edits as training data for sentence compression. They make the simplifying assumption that all selected edits retain the core meaning. Zanzotto and Pennacchiotti (2010) use edits as training data for textual entailment recognition. In addition to manually labeled edits, they use Wikipedia user comments and a co-training approach to leverage unlabeled edits. Woodsend and Lapata (2011) and Yatskar et al. (2010) use Wikipedia comments to identify relevant edits for learning sentence simplification.

The work by Max and Wisniewski (2010) is closely related to the approach proposed in this paper. They extract a corpus of rewritings, distinguish between weak semantic differences and strong semantic differences, and present a typology of multiple subclasses. Spelling corrections are heuristically identified but the task of automatic classification is deferred. Follow-up work by Dutrey et al. (2011) focuses on automatic paraphrase identification using a rule based approach and manually annotated examples.

Wikipedia vandalism detection is a user edits classification problem addressed by a yearly competition (since 2010) in conjunction with the CLEF conference (Potthast et al., 2010; Potthast and Holfeld, 2011). State-of-the-art solutions involve supervised machine learning using various content and metadata features. Content features use spelling, grammar, character- and word-level attributes. Many of them are relevant for our approach. Metadata features allow detection by patterns of usage, time and place, which are generally useful for the detection of online malicious activities (West et al., 2010; West and Lee, 2011). We deliberately refrain from using such features.

A wide range of methods and approaches has been applied to the similar tasks of textual entailment and paraphrase recognition, see Androutsopoulos and Malakasiotis (2010) for a comprehensive review. These are all related because paraphrases and bidirectional entailments represent types of fluency edits.

A different line of research uses classifiers to predict sentence-level fluency (Zwarts and Dras, 2008; Chae and Nenkova, 2009). These could be useful for fluency edits detection. Alternatively, user edits could be a potential source of human-produced training data for fluency models.

3 Definition of User Edits Scope

Within our approach we distinguish between *edit segments*, which represent the comparison (diff) between two document revisions, and *user edits*, which are the input for classification.

An *edit segment* is a contiguous sequence of deleted, inserted or equal words. The difference between two document revisions (v_i, v_j) is represented by a sequence of edit segments E. Each edit segment $(\delta, w_1^m) \in E$ is a pair, where $\delta \in \{deleted, inserted, equal\}$ and w_1^m is a m-word substring of v_i, v_j or both (respectively).

A *user edit* is a minimal set of sentences overlapping with deleted or inserted segments. Given the two sets of revision sentences (S_{v_i}, S_{v_j}), let

$$\phi(\delta, w_1^m) = \{s \in S_{v_i} \cup S_{v_j} \mid w_1^m \cap s \neq \emptyset\} \quad (1)$$

be the subset of sentences overlapping with a given edit segment, and let

$$\psi(s) = \{(\delta, w_1^m) \in E \mid w_1^m \cap s \neq \emptyset\} \quad (2)$$

be the subset of edit segments overlapping with a given sentence.

A user edit is a pair $(pre \subseteq S_{v_i}, post \subseteq S_{v_j})$ where

$$\forall s \in pre \cup post \;\; \forall \delta \in \{deleted, inserted\} \;\; \forall w_1^m$$
$$(\delta, w_1^m) \in \psi(s) \rightarrow \phi(\delta, w_1^m) \subseteq pre \cup post \quad (3)$$
$$\exists s \in pre \cup post \;\; \exists \delta \in \{deleted, inserted\} \;\; \exists w_1^m$$
$$(\delta, w_1^m) \in \psi(s) \quad (4)$$

Table 1 illustrates different types of edit segments and user edits. The term *replaced segment* refers to adjacent deleted and inserted segments. Example (1) contains a replaced segment because the deleted segment ("*1700s*") is adjacent to the inserted segment ("*18th century*"). Example (2) contains an inserted segment ("*and largest professional*"), a replaced segment ("*(est.*" → "*established in*") and a deleted segment ("*)*"). User edits of both examples consist of a single *pre* sentence and a single *post* sentence because deleted and inserted segments do not cross any sentence boundary. Example (3) contains a replaced segment (". *He*" → "*who*"). In this case the deleted segment (". *He*") overlaps with two sentences and therefore the user edit consists of two *pre* sentences.

(1) Revisions 368209202 & 378822230	
pre	("By the mid **1700s**, Medzhybizh was the seat of power in Podilia Province.")
post	("By the mid **18th century**, Medzhybizh was the seat of power in Podilia Province.")
diff	(equal , "By the mid") , (**deleted**, "1700s") , (**inserted** , "18th century") , (equal , ", Medzhybizh was the seat of power in Podilia Province.")
(2) Revisions 148109085 & 149440273	
pre	("Original Society of Teachers of the Alexander Technique **(est.** 1958**)**.")
post	("Original **and largest professional** Society of Teachers of the Alexander Technique **established in** 1958.")
diff	(equal , "Original") , (**inserted** , "and largest professional") , (equal , "Society of Teachers of the Alexander Technique") , (**deleted** , "(est.") , (**inserted** , " established in") , (equal , "1958") , (**deleted** , ")") , (equal , ".")
(3) Revisions 61406809 & 61746002	
pre	("Fredrik Modin is a Swedish ice hockey left winger**.**" , "**He** is known for having one of the hardest slap shots in the NHL.")
post	("Fredrik Modin is a Swedish ice hockey left winger **who** is known for having one of the hardest slap shots in the NHL.")
diff	(equal , "Fredrik Modin is a Swedish ice hockey left winger") , (**deleted** , ". He") , (**inserted** , "who") , (equal , "is known for having one of the hardest slap shots in the NHL.")

Table 1: Examples of user edits and the corresponding edit segments (revision numbers correspond to the English Wikipedia).

4 Features for Edits Classification

We design a set of features for supervised classification of user edits. The design is guided by two main considerations: simplicity and interoperability. Simplicity is important because there are potentially hundreds of millions of user edits to be classified. This amount continues to grow at rapid pace and a scalable solution is required. Interoperability is important because millions of user edits are available in multiple languages. Wikipedia is a flagship project, but there are other collaborative editing projects. The solution should preferably be language- and project-independent. Consequently, we refrain from deeper syntactic parsing, Wikipedia-specific features, and language resources that are limited to English.

Our basic intuition is that longer edits are likely to be factual and shorter edits are likely to be fluency edits. The **baseline method** is therefore character-level edit distance (Levenshtein, 1966) between pre- and post-edited text.

Six feature categories are added to the baseline. Most features take the form of threefold counts referring to deleted, inserted and equal elements of each user edit. For instance, example (1) in Table 1 has one deleted token, two inserted tokens and 14 equal tokens. Many features use string similarity calculated over alternative representations.

Character-level features include counts of deleted, inserted and equal characters of different types, such as word & non-word characters or digits & non-digits. Character types may help identify edits types. For example, the change of digits may suggest a factual edit while the change of non-word characters may suggest a fluency edit.

Word-level features count deleted, inserted and equal words using three parallel representations: original case, lower case, and lemmas. Word-level edit distance is calculated for each representation. Table 2 illustrates how edit distance may vary across different representations.

Rep.	*User Edit*		*Dist*
Words	pre	Branch lines were built in Kenya	4
	post	A branch line was built in Kenya	
Lowcase	pre	branch lines were built in kenya	3
	post	a branch line was built in kenya	
Lemmas	pre	branch line be build in Kenya	1
	post	a branch line be build in Kenya	
PoS tags	pre	NN NNS VBD VBN IN NNP	2
	post	DT NN NN VBD VBN IN NNP	
NE tags	pre	LOCATION	0
	post	LOCATION	

Table 2: Word- and tag-level edit distance measured over different representations (example from Wikipedia revisions 2678278 & 2682972).

Fluency edits may shift words, which sometimes may be slightly modified. Fluency edits may also add or remove words that already appear in context. Optimal calculation of edit distance with shifts is computationally expensive (Shapira and Storer, 2002). Translation error rate (TER) provides an approximation but it is designed for the needs of machine translation evaluation (Snover et al., 2006). To have a more sensitive estimation of the degree of edit, we compute the minimal character-level edit distance between every pair of words that belong to different edit segments. For each pair of edit segments $(\delta, w_1^m), (\delta', w'^k_1)$ overlapping with a user edit, if $\delta \neq \delta'$ we compute:

$$\forall w \in w_1^m : \min_{w' \in w'^k_1} EditDist(w, w') \quad (5)$$

Binned counts of the number of words with a minimal edit distance of 0, 1, 2, 3 or more characters are accumulated per edit segment type (equal, deleted or inserted).

Part-of-speech (PoS) features include counts of deleted, inserted and equal PoS tags (per tag) and edit distance at the tag level between PoS tags before and after the edit. Similarly, **named-entity (NE) features** include counts of deleted, inserted and equal NE tags (per tag, excluding *OTHER*) and edit distance at the tag level between NE tags before and after the edit. Table 2 illustrates the edit distance at different levels of representation. We assume that a deleted NE tag, e.g. *PERSON* or *LOCATION*, could indicate a factual edit. It could however be a fluency edit where the NE is replaced by a co-referent like "*she*" or "*it*". Even if we encounter an inserted *PRP* PoS tag, the features do not capture the explicit relation between the deleted NE tag and the inserted PoS tag. This is an inherent weakness of these features when compared to parsing-based alternatives.

An additional set of counts, NE values, describes the number of deleted, inserted and equal normalized values of numeric entities such as numbers and dates. For instance, if the word "100" is replaced by "200" and the respective numeric values 100.0 and 200.0 are normalized, the counts of deleted and inserted NE values will be incremented and suggest a factual edit. If on the other hand "100" is replaced by "hundred" and the latter is normalized as having the numeric value 100.0, then the count of equal NE values will be incremented, rather suggesting a fluency edit.

Acronym features count deleted, inserted and equal acronyms. Potential acronyms are extracted from word sequences that start with a capital letter and from words that contain multiple capital letters. If, for example, "UN" is replaced by "United Nations", "MicroSoft" by "MS" or "Jean Pierre" by "J.P", the count of equal acronyms will be incremented, suggesting a fluency edit.

The last category, **language model (LM) features**, takes a different approach. These features look at n-gram based sentence probabilities before and after the edit, with and without normalization with respect to sentence lengths. The ratio of the two probabilities, $\hat{P}_{ratio}(pre, post)$ is computed as follows:

$$\hat{P}(w_1^m) \approx \prod_{i=1}^{m} P(w_i | w_{i-n+1}^{i-1}) \quad (6)$$

$$\hat{P}_{norm}(w_1^m) = \hat{P}(w_1^m)^{\frac{1}{m}} \quad (7)$$

$$\hat{P}_{ratio}(pre, post) = \frac{\hat{P}_{norm}(post)}{\hat{P}_{norm}(pre)} \quad (8)$$

$$\begin{aligned} \log \hat{P}_{ratio}(pre, post) &= \log \frac{\hat{P}_{norm}(post)}{\hat{P}_{norm}(pre)} \quad (9) \\ &= \log \hat{P}_{norm}(post) - \log \hat{P}_{norm}(pre) \\ &= \frac{1}{|post|} \log \hat{P}(post) - \frac{1}{|pre|} \log \hat{P}(pre) \end{aligned}$$

Where $\hat{P}$ is the sentence probability estimated as a product of n-gram conditional probabilities and $\hat{P}_{norm}$ is the sentence probability normalized by the sentence length. We hypothesize that the relative change of normalized sentence probabilities is related to the edit type. As an additional feature, the number of out of vocabulary (OOV) words before and after the edit is computed. The intuition

	Dataset	*Labeled Subset*
Number of User Edits:		
	923,820 (100%)	2,008 (100%)
Edit Segments Distribution:		
Replaced	535,402 (57.96%)	1,259 (62.70%)
Inserted	235,968 (25.54%)	471 (23.46%)
Deleted	152,450 (16.5%)	278 (13.84%)
Character-level Edit Distance Distribution:		
1	202,882 (21.96%)	466 (23.21%)
2	81,388 (8.81%)	198 (9.86%)
3-10	296,841 (32.13%)	645 (32.12%)
11-100	342,709 (37.10%)	699 (34.81%)
Word-level Edit Distance Distribution:		
1	493,095 (53.38%)	1,008 (54.18%)
2	182,770 (19.78%)	402 (20.02%)
3	77,603 (8.40%)	161 (8.02%)
4-10	170,352 (18.44%)	357 (17.78%)
Labels Distribution:		
Fluency	-	1,008 (50.2%)
Factual	-	1,000 (49.8%)

Table 3: Dataset of nearly 1 million user edits with single deleted, inserted or replaced segments, of which 2K are labeled. The labels are almost equally distributed. The distribution over edit segment types and edit distance intervals is detailed.

is that unknown words are more likely to be indicative of factual edits.

5 Experiments

5.1 Experimental Setup

First, we extract a large amount of user edits from revision histories of the English Wikipedia.[3] The extraction process scans pairs of subsequent revisions of article pages and ignores any revision that was reverted due to vandalism. It parses the Wikitext and filters out markup, hyperlinks, tables and templates. The process analyzes the clean text of the two revisions[4] and computes the difference between them.[5] The process identifies the overlap between edit segments and sentence boundaries and extracts user edits. Features are calculated and user edits are stored and indexed. LM features are calculated against a large English 4-gram language model built by SRILM (Stolcke, 2002) with modified interpolated Kneser-Ney smoothing using the AFP and Xinhua portions of the English Gigaword corpus (LDC2003T05).

We extract a total of 4.3 million user edits of which 2.52 million (almost 60%) are insertions and deletions of complete sentences. Although these may include fluency edits such as sentence reordering or rewriting from scratch, we assume that the large majority is factual. Of the remaining 1.78 million edits, the majority (64.5%) contains single deleted, inserted or replaced segments. We decide to focus on this subset because sentences with multiple non-contiguous edit segments are more likely to contain mixed cases of unrelated factual and fluency edits, as illustrated by example (2) in Table 1. Learning to classify contiguous edit segments seems to be a reasonable way of breaking down the problem into smaller parts. We filter out user edits with edit distance longer than 100 characters or 10 words that we assume to be factual. The resulting dataset contains 923,820 user edits: 58% replaced segments, 25.5% inserted segments and 16.5% deleted segments.

Manual labeling of user edits is carried out by a group of annotators with near native or native level of English. All annotators receive the same written guidelines. In short, fluency labels are assigned to edits of letter case, spelling, grammar, synonyms, paraphrases, co-referents, language and style. Factual labels are assigned to edits of dates, numbers and figures, named entities, semantic change or disambiguation, addition or removal of content. A random set of 2,676 instances is labeled: 2,008 instances with a majority agreement of at least two annotators are selected as training set, 270 instances are held out as development set, 164 trivial fluency corrections of a single letter's case and 234 instances with no clear agreement among annotators are excluded. The last group (8.7%) emphasizes that the task is, to a limited extent, subjective. It suggests that automated classification of certain user edits would be difficult. Nevertheless, inter-rater agreement between annotators is high to very high. Kappa values between 0.74 to 0.84 are measured between six pairs of annotators, each pair annotated a common subset of at least 100 instances. Table 3 describes the resulting dataset, which we also make available to the research community.[6]

[3] Dump of all pages with complete edit history as of January 15, 2011 (342GB bz2), http://dumps.wikimedia.org.

[4] Tokenization, sentence split, PoS & NE tags by Stanford CoreNLP, http://nlp.stanford.edu/software/corenlp.shtml.

[5] Myers' $O(ND)$ difference algorithm (Myers, 1986), http://code.google.com/p/google-diff-match-patch.

[6] Available for download at `http://staff.`

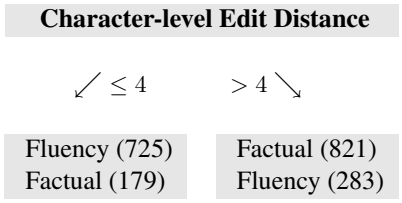

Figure 1: A decision tree that uses character-level edit distance as a sole feature. The tree correctly classifies 76% of the labeled user edits.

Feature set	**SVM**	**RF**	**Logit**
Baseline	76.26%	76.26%	76.34%
+ Char-level	83.71%†	84.45%†	84.01%†
+ Word-level	78.38%$^{\dagger\vee}$	81.38%$^{\dagger\wedge}$	78.13%$^{\dagger\vee}$
+ PoS	76.58%$^{\vee}$	76.97%	78.35%$^{\dagger\wedge}$
+ NE	82.71%†	83.12%†	82.38%†
+ Acronyms	76.55%	76.61%	76.96%
+ LM	76.20%	77.42%	76.52%
All Features	**87.14%**$^{\dagger\wedge}$	**87.14%**†	**85.64%**$^{\dagger\vee}$

Table 4: Classification accuracy using the baseline, each feature set added to the baseline, and all features combined. Statistical significance at $p < 0.05$ is indicated by † w.r.t the baseline (using the same classifier), and by $^{\wedge}$ w.r.t to another classifier marked by $^{\vee}$ (using the same features). Highest accuracy per classifier is marked in bold.

5.2 Feature Analysis

We experiment with three classifiers: Support Vector Machines (SVM), Random Forests (RF) and Logistic Regression (Logit).[7] SVMs (Cortes and Vapnik, 1995) and Logistic Regression (or Maximum Entropy classifiers) are two widely used machine learning techniques. SVMs have been applied to many text classification problems (Joachims, 1998). Maximum Entropy classifiers have been applied to the similar tasks of paraphrase recognition (Malakasiotis, 2009) and textual entailment (Hickl et al., 2006). Random Forests (Breiman, 2001) as well as other decision tree algorithms are successfully used for classifying Wikipedia edits for the purpose of vandalism detection (Potthast et al., 2010; Potthast and Holfeld, 2011).

Experiments begin with the edit-distance baseline. Then each one of the feature groups is separately added to the baseline. Finally, all features are evaluated together. Table 4 reports the percentage of correctly classified edits (classifiers' accuracy), and Table 5 reports the fraction of correctly classified edits per type. All results are for 10-fold cross validation. Statistical significance against the baseline and between classifiers is calculated at $p < 0.05$ using paired t-test.

Feature set	**SVM** *flu. / fac.*	**RF** *flu. / fac.*	**Logit** *flu. / fac.*
Baseline	0.85 / 0.67	0.74 / 0.79	0.85 / 0.67
+ Char-level	0.85 / 0.82	0.83 / 0.86	0.86 / 0.82
+ Word-level	0.88 / 0.69	0.81 / 0.82	0.86 / 0.70
+ PoS	0.85 / 0.68	0.78 / 0.76	0.84 / 0.72
+ NE	0.86 / 0.79	0.79 / 0.87	0.87 / 0.78
+ Acronyms	0.87 / 0.66	0.83 / 0.70	0.86 / 0.68
+ LM	0.85 / 0.67	0.79 / 0.76	0.84 / 0.69
All Features	0.88 / 0.86	0.86 / 0.88	0.87 / 0.84

Table 5: Fraction of correctly classified edits per type: fluency edits (left) and factual edits (right), using the baseline, each feature set added to the baseline, and all features combined.

The first interesting result is the highly predictive power of the single-feature baseline. It confirms the intuition that longer edits are mainly factual. Figure 1 shows that the edit distance of 72% of the user edits labeled as fluency is between 1 to 4, while the edit distance of 82% of those labeled as factual is greater than 4. The cut-off value is found by a single-node decision tree that uses edit distance as a sole feature. The tree correctly classifies 76% of the instances. This result implies that the actual challenge is to correctly classify short factual edits and long fluency edits.

Character-level features and named-entity features lead to significant improvements over the baseline for all classifiers. Their strength lies in their ability to identify short factual edits such as changes of numeric values or proper names. Word-level features also significantly improve the baseline but their contribution is smaller. PoS and acronym features lead to small statistically-insignificant improvements over the baseline.

The poor contribution of LM features is surprising. It might be due to the limited context of n-grams, but it might be that LM probabilities are not a good predictor for the task. Removing LM features from the set of all features

science.uva.nl/~abronner/uec/data.

[7] Using Weka classifiers: SMO (SVM), RandomForest & Logistic (Hall et al., 2009). Classifier's parameters are tuned using the held-out development set.

Fluency Edits Misclassified as Factual	
Equivalent or redundant in context	14
Paraphrases	13
Equivalent numeric patterns	7
Replacing first name with last name	4
Acronyms	4
Non specific adjectives or adverbs	3
Other	5

Factual Edits Misclassified as Fluency	
Short correction of content	35
Opposites	3
Similar names	3
Noise (unfiltered vandalism)	3
Other	6

Table 6: Error types based on manual examination of 50 fluency edit misclassifications and 50 factual edit misclassifications.

Correctly Classified Fluency Edits
"Adventure education ~~makes intentional use of~~ **intentionally uses** challenging experiences for learning."
"He served as president from October 1 , 1985 ~~and retired~~ **through his retirement** on June 30 , 2002."
"In 1973, he ~~helped organize~~ **assisted in organizing** his first ever visit to the West."

Correctly Classified Factual Edits
"Over the course of the next ~~two years~~ **five months**, the unit completed a series of daring raids."
"**Scottish born** David Tennant has reportedly said he would like his Doctor to wear a kilt."
"This family joined the strip ~~in late 1990~~ **around March 1991**."

Table 7: Examples of correctly classified user edits. Deleted segments are struck out, inserted are bold (revision numbers are omitted for brevity).

leads to a small decrease in classification accuracy, namely 86.68% instead of 87.14% for SVM. This decrease is not statistically significant.

The highest accuracy is achieved by both SVM and RF and there are few significant differences among the three classifiers. The fraction of correctly classified edits per type (Table 5) reveals that for SVM and Logit, most fluency edits are correctly classified by the baseline and most improvements over the baseline are attributed to better classification of factual edits. This is not the case for RF, where the fraction of correctly classified factual edits is higher and the fraction of correctly classified fluency edits is lower. This insight motivates further experimentation. Repeating the experiment with a meta-classifier that uses a majority voting scheme, achieves an improved accuracy of 87.58%. This improvement is not statistically significant.

5.3 Error Analysis

To have better understanding of errors made by the classifier, 50 fluency edit misclassifications and 50 factual edit misclassifications are randomly selected and manually examined. The errors are grouped into categories as summarized in Table 6. These explain certain limitations of the classifier and suggest possible improvements.

Fluency edit misclassifications: 14 instances (28%) are phrases (often co-referents) that are either equivalent or redundant in the given context. For example: "*in 1986*" → "*that year*", "*when she returned*" → "*when Ruffa returned*" and "*the core member of the group are*" → "*the core members are*". 13 (26%) are paraphrases misclassified as factual edits. Examples are: "*made cartoons*" → "*produced animated cartoons*" and "*with the implication that they are similar to*" → "*implying a connection to*". 7 modify numeric patterns that do not change the meaning such as the year "*37*" → "*1937*". 4 replace a first name of a person with the last name. 4 contain acronyms, e.g. "*Display PostScript*" → "*Display PostScript (or DPS)*". Acronym features are correctly identified but the classifier fails to recognize a fluency edit. 3 modify adjectives or adverbs that do not change the meaning such as "*entirely*" and "*various*".

Factual edit misclassifications: the big majority, 35 instances (70%), could be characterized as short corrections, often replacing a similar word, that make the content more accurate or more precise. Examples (context is omitted): "*city*" → "*village*", "*emigrated*" → "*immigrated*" and "*electrical*" → "*electromagnetic*". 3 are opposites or antonyms such as "*previous*" → "*next*" and "*lived*" → "*died*". 3 are modifications of similar person or entity names, e.g. "*Kelly*" → "*Kate*". 3 are instances of unfiltered vandalism, i.e. noisy examples. Other misclassifications include verb tense modifications such as "*is*" → "*was*" and "*consists*" → "*consisted*". These are difficult to

Comment	Test Set Size	Classified as Fluency Edits
"*grammar*"	1,122	88.9%
"*spelling*"	2,893	97.6%
"*typo*"	3,382	91.6%
"*copyedit*"	3,437	68.4%
Random set	5,000	49.4%

Table 8: Classifying unlabeled data selected by user comments that suggest a fluency edit. The SVM classifier is trained using the labeled data. User comments are not used as features.

classify because the modification of verb tense in a given context is sometimes factual and sometimes a fluency edit.

These findings agree with the feature analysis. Fluency edit misclassifications are typically longer phrases that carry the same meaning while factual edit misclassifications are typically single words or short phrases that carry different meaning. The main conclusion is that the classifier should take into account explicit content and context. Putting aside the consideration of simplicity and interoperability, features based on co-reference resolution and paraphrase recognition are likely to improve fluency edits classification, and features from language resources that describe synonymy and antonymy relations are likely to improve factual edits classification. While this conclusion may come at no surprise, it is important to highlight the high classification accuracy that is achieved without such capabilities and resources. Table 7 presents several examples of correct classification produced by our classifier.

6 Exploiting Unlabeled Data

We extracted a large set of user edits but our approach has been limited to a restricted number of labeled examples. This section attempts to find whether the classifier generalizes beyond labeled data and whether unlabeled data could be used to improve classification accuracy.

6.1 Generalizing Beyond Labeled Data

The aim of the next experiment is to test how well the supervised classifier generalizes beyond the labeled test set. The problem is the availability of test data. There is no shared task for user edits classification and no common test set to evaluate against. We resort to Wikipedia user comments. It is a problematic option because it is unreliable. Users may add a comment when submitting an edit, but it is not mandatory. The comment is a free text with no predefined structure. It could be meaningful or nonsense. The comment is per revision. It may refer to one, some or all edits submitted for a given revision. Nevertheless, we identify several keywords that represent certain types of fluency edits: "*grammar*", "*spelling*", "*typo*", and "*copyedit*". The first three clearly indicate grammar and spelling corrections. The last indicates a correction of format and style, but also of accuracy of the text. Therefore it only represents a bias towards fluency edits.

We extract unlabeled edits whose comment is equal to one of the keywords and construct a test set per keyword. An additional test set consists of randomly selected unlabeled edits with any comment. The five test sets are classified by the SVM classifier trained using the labeled data and the set of all features. To remove any doubt, user comments are not part of any feature of the classifier.

The results in Table 8 show that most unlabeled edits whose comments are "*grammar*", "*spelling*" or "*typo*" are indeed classified as fluency edits. The classification of edits whose comment is "*copyedit*" is biased towards fluency edits, but as expected the result is less distinct. The classification of the random set is balanced, as expected.

Replaced by	Frequency	Edit class
"*second*"	144	Factual
"*First*"	38	Fluency
"*last*"	31	Factual
"*1st*"	22	Fluency
"*third*"	22	Factaul

Table 9: User edits replacing the word "*first*" with another single word: most frequent 5 out of 524.

Replaced by	Frequency	Replaced by	Frequency
"*Adams*"	7	"*Squidward*"	6
"*Joseph*"	7	"*Alexander*"	5
"*Einstein*"	6	"*Davids*"	5
"*Galland*"	6	"*Haim*"	5
"*Lowe*"	6	"*Hickes*"	5

Table 10: Fluency edits replacing the word "*He*" with proper noun: most frequent 10 out of 1,381.

Feature set	**SVM**	**RF**	**Logit**
Baseline	76.26%	76.26%	76.34%
All Features	87.14%$^{\dagger\wedge}$	87.14%†	85.64%$^{\dagger\vee}$
Unlabeled only	78.11%$^{\vee}$	83.49%$^{\dagger\wedge}$	78.78%$^{\dagger\vee}$
Base + unlabeled	80.86%$^{\dagger\vee}$	85.45%$^{\dagger\wedge}$	81.83%$^{\dagger\vee}$
All + unlabeled	87.23%	**88.35%**$^{\ddagger\dagger\wedge}$	85.92%$^{\vee}$

Table 11: Classification accuracy using features from unlabeled data. The first two rows are identical to Table 4. Statistical significance at $p < 0.05$ is indicated by: † w.r.t the baseline; ‡ w.r.t all features excluding features from unlabeled data; and $^{\wedge}$ w.r.t to another classifier marked by $^{\vee}$ (using the same features). The best result is marked in bold.

6.2 Features from Unlabeled Data

The purpose of the last experiment is to exploit unlabeled data in order to extract additional features for the classifier. The underlying assumption is that reoccurring patterns may indicate whether a user edit is factual or a fluency edit.

We could assume that fluency edits would reoccur across many revisions, while factual edits would only appear in revisions of specific documents. However, this assumption does not necessarily hold. Table 9 gives a simple example of single word replacements for which the most reoccurring edit is actually factual and other factual and fluency edits reoccur in similar frequencies.

Finding user edits reoccurrence is not trivial. We could rely on exact matches of surface forms, but this may lead to data sparseness issues. Fluency edits that exchange co-referents and proper nouns, as illustrated by the example in Table 10, may reoccur frequently but this fact could not be revealed by exact matching of specific proper nouns. On the other hand, using a bag of word approach may find too many unrelated edits.

We introduce a two-step method that measures the reoccurrence of edits in unlabeled data using exact and approximate matching over multiple representations. The method provides a set of frequencies that is fed into the classifier and allows for learning subtle patterns of reoccurrence. Staying consistent with our initial design considerations, the method is simple and interoperable.

Given a user edit $(pre, post)$, the method does not compare pre with $post$ in any way. It only compares pre with pre-edited sentences of other unlabeled edits and $post$ with post-edited sentences of other unlabeled edits. The first step is to select candidates using a bag of words approach. The second step is a comparison of the user edit with each one of the candidates while incrementing counts of similarity measures. These account for exact matches between different representations (original and low case, lemmas, PoS and NE tags) as well as for approximate matches using character- and word-level edit distance between those representations. An additional feature is the number of distinct documents in the candidate set.

We compute the set of features for the labeled dataset based on the unlabeled data. The number of candidates is set to 1,000 per user edit. We re-train the classifiers using five configurations: *Baseline* and *All Features* are identical to the first experiment. *Unlabeled only* uses the new feature set without any other feature. *Base + Unlabeled* adds the new feature set to the baseline. *All + Unlabeled* uses all available features. All results are for 10-fold cross validation with statistical significance at $p < 0.05$ by paired t-test, see Table 11.

We find that features extracted from unlabeled data outperform the baseline and lead to statistically significant improvements when added to it. The combination of all features allows Random Forests to achieve the highest statistically significant accuracy level of 88.35%.

7 Conclusions

This work addresses the task of user edits classification as factual or fluency edits. It adopts a supervised machine learning approach and uses character- and word- level features, part-of-speech tags, named entities, language model probabilities, and a set of features extracted from large amounts of unlabeled data. Our experiments with contiguous user edits extracted from revision histories of the English Wikipedia achieve high classification accuracy and demonstrate generalization to data beyond labeled edits.

Our approach shows that machine learning techniques can successfully distinguish between user edit types, making them a favorable alternative to heuristic solutions. The simple and adaptive nature of our method allows for application to large and evolving sets of user edits.

Acknowledgments. This research was funded in part by the European Commission through the CoSyne project FP7-ICT-4-248531.

References

A. Aji, Y. Wang, E. Agichtein, and E. Gabrilovich. 2010. Using the past to score the present: Extending term weighting models through revision history analysis. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 629–638.

I. Androutsopoulos and P. Malakasiotis. 2010. A survey of paraphrasing and textual entailment methods. *Journal of Artificial Intelligence Research*, 38(1):135–187.

L. Breiman. 2001. Random forests. *Machine learning*, 45(1):5–32.

J. Chae and A. Nenkova. 2009. Predicting the fluency of text with shallow structural features: case studies of machine translation and human-written text. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 139–147.

C. Cortes and V. Vapnik. 1995. Support-vector networks. *Machine learning*, 20(3):273–297.

C. Dutrey, D. Bernhard, H. Bouamor, and A. Max. 2011. Local modifications and paraphrases in Wikipedia's revision history. *Procesamiento del Lenguaje Natural*, Revista no 46:51–58.

M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I.H. Witten. 2009. The WEKA data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18.

A. Hickl, J. Williams, J. Bensley, K. Roberts, B. Rink, and Y. Shi. 2006. Recognizing textual entailment with LCCs GROUNDHOG system. In *Proceedings of the Second PASCAL Challenges Workshop*.

T. Joachims. 1998. Text categorization with support vector machines: Learning with many relevant features. *Machine Learning: ECML-98*, pages 137–142.

A. Kittur, B. Suh, B.A. Pendleton, and E.H. Chi. 2007. He says, she says: Conflict and coordination in Wikipedia. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 453–462.

V.I. Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8):707–710.

P. Malakasiotis. 2009. Paraphrase recognition using machine learning to combine similarity measures. In *Proceedings of the ACL-IJCNLP 2009 Student Research Workshop*, pages 27–35.

A. Max and G. Wisniewski. 2010. Mining naturally-occurring corrections and paraphrases from Wikipedia's revision history. In *Proceedings of LREC*, pages 3143–3148.

E.W. Myers. 1986. An $O(ND)$ difference algorithm and its variations. *Algorithmica*, 1(1):251–266.

R. Nelken and E. Yamangil. 2008. Mining Wikipedia's article revision history for training computational linguistics algorithms. In *Proceedings of the AAAI Workshop on Wikipedia and Artificial Intelligence: An Evolving Synergy*, pages 31–36.

S. Nunes, C. Ribeiro, and G. David. 2011. Term weighting based on document revision history. *Journal of the American Society for Information Science and Technology*, 62(12):2471–2478.

M. Potthast and T. Holfeld. 2011. Overview of the 2nd international competition on Wikipedia vandalism detection. *Notebook for PAN at CLEF 2011*.

M. Potthast, B. Stein, and T. Holfeld. 2010. Overview of the 1st international competition on Wikipedia vandalism detection. *Notebook Papers of CLEF*, pages 22–23.

D. Shapira and J. Storer. 2002. Edit distance with move operations. In *Combinatorial Pattern Matching*, pages 85–98.

M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of Association for Machine Translation in the Americas*, pages 223–231.

A. Stolcke. 2002. SRILM-an extensible language modeling toolkit. In *Proceedings of the international conference on spoken language processing*, volume 2, pages 901–904.

F.B. Viegas, M. Wattenberg, and K. Dave. 2004. Studying cooperation and conflict between authors with history flow visualizations. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 575–582.

A.G. West and I. Lee. 2011. Multilingual vandalism detection using language-independent & ex post facto evidence. *Notebook for PAN at CLEF 2011*.

A.G. West, S. Kannan, and I. Lee. 2010. Detecting Wikipedia vandalism via spatio-temporal analysis of revision metadata. In *Proceedings of the Third European Workshop on System Security*, pages 22–28.

K. Woodsend and M. Lapata. 2011. Learning to simplify sentences with quasi-synchronous grammar and integer programming. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 409–420.

E. Yamangil and R. Nelken. 2008. Mining Wikipedia revision histories for improving sentence compression. In *Proceedings of ACL-08: HLT, Short Papers*, pages 137–140.

M. Yatskar, B. Pang, C. Danescu-Niculescu-Mizil, and L. Lee. 2010. For the sake of simplicity: Unsupervised extraction of lexical simplifications from Wikipedia. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 365–368.

F.M. Zanzotto and M. Pennacchiotti. 2010. Expanding textual entailment corpora from Wikipedia using co-training. In *Proceedings of the 2nd Workshop on Collaboratively Constructed Semantic Resources, COLING 2010*.

S. Zwarts and M. Dras. 2008. Choosing the right translation: A syntactically informed classification approach. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 1153–1160.