Four types of context for automatic spelling correction

Michael Flor

Educational Testing Service
Rosedale Road
Princeton, NJ, 08541, USA
mflor@ets.org

ABSTRACT. *This paper presents an investigation on using four types of contextual information for improving the accuracy of automatic correction of single-token non-word misspellings. The task is framed as contextually-informed re-ranking of correction candidates. Immediate local context is captured by word n-grams statistics from a Web-scale language model. The second approach measures how well a candidate correction fits in the semantic fabric of the local lexical neighborhood, using a very large Distributional Semantic Model. In the third approach, recognizing a misspelling as an instance of a recurring word can be useful for re-ranking. The fourth approach looks at context beyond the text itself. If the approximate topic can be known in advance, spelling correction can be biased towards the topic. Effectiveness of proposed methods is demonstrated with an annotated corpus of 3,000 student essays from international high-stakes English language assessments. The paper also describes an implemented system that achieves high accuracy on this task.*

RÉSUMÉ. *Cet article présente une enquête sur l'utilisation de quatre types d'informations contextuelles pour améliorer la précision de la correction automatique de fautes d'orthographe de mots seuls. La tâche est présentée comme un reclassement contextuellement informé. Le contexte local immédiat, capturé par statistique de mot n-grammes est modélisé à partir d'un modèle de langage à l'échelle du Web. La deuxième méthode consiste à mesurer à quel point une correction s'inscrit dans le tissu sémantique local, en utilisant un très grand modèle sémantique distributionnel. La troisième approche reconnaissant une faute d'orthographe comme une instance d'un mot récurrent peut être utile pour le reclassement. La quatrième approche s'attache au contexte au-delà du texte lui-même. Si le sujet approximatif peut être connu à l'avance, la correction orthographique peut être biaisée par rapport au sujet. L'efficacité des méthodes proposées est démontrée avec un corpus annoté de 3 000 travaux d'étudiants des évaluations internationales de langue anglaise. Le document décrit également un système mis en place qui permet d'obtenir une grande précision sur cette tâche.*

KEY WORDS: *automatic spelling correction, context, n-grams, language models.*

MOTS-CLÉS : *correction automatique de l'orthographe, contexte, n-grammes.*

TAL. Volume 53 – n° 3/2012, pages 61 à 99

1. Introduction

Misspellings are pervasive. They are found in all kinds of writing, including in student essays. Lunsford and Lunsford (2008) found that spelling errors constituted about 6.5% of all types of errors found in a US national sample of college composition essays, despite the fact that writers had access to spellcheckers. Desmet and Balthazor (2005) found that spelling errors are among the five most frequent errors in first-year college composition of US students. Misspellings are even more ubiquitous in texts written by non-native speakers of English, especially English language learners (ELL) (Flor and Futagi, 2012). The types of misspellings produced by ELL writers are typically different from errors produced by native speakers (Hovermale, 2010; Okada 2005; Cook, 1997).

In the area of automatic assessment of writing, detection of misspellings is utilized in computer-aided language learning applications and in automatic scoring systems, especially when feedback to users is involved (Dikli, 2006; Warschauer and Ware, 2006) – both for aggregate evaluations and for specific feedback on spelling errors (ETS, 2007). Yet spelling errors may have a deeper influence on automated text assessment. As noted by Nagata *et al*. (2011), sub-optimal automatic detection of grammar and mechanics errors may be attributed to poor performance of NLP tools over noisy text.

Presence of spelling errors also hinders systems that require only lexical analysis of text (Landauer *et al*., 2003; Pérez *et al*., 2004). Granger and Wynne (1999) have shown that spelling errors can affect automated estimates of lexical variation, such as type/token ratio, which in turn are used as predictors of text quality (Yu, 2010; Crossley *et al*., 2008). In the context of automated preposition and determiner error correction in L2 English, De Felice and Pulman (2008) noted that the process is often disrupted by misspellings. Futagi (2010) described how misspellings pose problems in development of a tool for detection of phraseological collocation errors. Rozovskaya *et al.* (2012) note that automatic correction of spelling errors enhances automatic correction of preposition and determiner errors made by non-native English speakers. Spelling errors are also problematic for automatic content scoring systems, where the focus is on evaluating the correctness of responses rather than their language quality (Sukkarieh and Blackmore, 2009; Leacock and Chodorow, 2003).

Classic approaches to the problem of spelling correction of non-word errors were reviewed by Kukich (1992) and Mitton (1996). Basically, a non-word misspelling is a string that is not found in dictionary. The standard approach for error detection is using good spelling dictionaries. The typical approach for correction of non-word errors is to include modules for computing edit distance (Damerau, 1964; Levenshtein, 1966) and phonetic similarity. These are used for ranking generated suggestions by their similarity to the misspelled word. A more recent feature utilizes word frequency data as an additional measure for candidate ranking. Mitton (2008)

and Deorowicz and Ciura (2005) described state-of-the-art approaches to non-word correction without contextual information. The use of noisy channel model for spelling correction was introduced by Kernighan *et al*. (1990). An early approach for using contextual data for non-word error correction was described by Brill and Moore (2000). The use of Google Web1T n-gram corpus (Brants and Franz, 2006) for context-informed spelling correction of real-word and simulated non-word errors was described by Carlson and Fette (2007). Use of text data from the Web for spelling correction was described by Whitelaw *et al*. (2009) and Chen *et al*. (2007).

This paper provides an outline of research exploring specific contextual influences for improving automatic correction of non-word misspellings. Consider a misspelling like *forst*. Candidate corrections could include *first*, *forest*, *frost*, and even *forced*. But which one is the right one? In a context like "*forst fires in Yellowstone*", *forest* is a likely candidate. For "*forst in line*", *first* seems more adequate. In a context like "*...ice crystals ... forst...*", *frost* is quite plausible. We present systematic ways to exploit such information. The rest of this paper is structured as follows. First, the corpus of texts used in this study is described in some detail, since the research was conducted with a focus on automatically correcting spelling errors in student essays. Then, the spelling correction system is described. We present the data on error detection, then baseline results for error correction without context. Next, four types of contexts and specific algorithms that use them are described: 1) *n*-grams – which candidate correction fits better in the sequence of words where misspelling is found; 2) word associations – which candidate correction has better semantic fit with the words around the misspelling; 3) word repetitions – words occurring multiple times in a text can help finding adequate corrections; 4) topical bias – correction candidates can be preferred by considering words that are especially relevant to the topic of the text. Results are presented for each type of context separately, and for combinations of the methods.

2. The corpus

The ETS Spelling Corpus is a collection of essays, annotated for misspellings by trained annotators. It is developed for evaluation of spellcheckers, and for research on patterns of misspellings produced by both native English speakers and English language learners.

The corpus comprises essays written by examinees on the writing sections of GRE® (Graduate Record Examinations) and TOEFL® (Test of English as a Foreign Language) (ETS, 2011a,b). The TOEFL test includes two different writing tasks. On the Independent task, examinees write a short opinion essay, on a pre-assigned topic. On the Integrated writing task, examinees write a summary essay that compares arguments from two different sources (both supplied during the test). The GRE Analytical Writing Section also includes two different writing tasks. On the GRE Issue task, test takers write a short argumentative essay by taking a position on an assigned topic. On the GRE Argument task, test takers read a short argument text

and then write an essay evaluating the soundness of the prompt argument. Both TOEFL and GRE are delivered on computer (at test centers around the world and via Internet), always using the standard English language computer keyboard (QWERTY). Editing tools such as a spellchecker are not provided in the test-delivery software (ETS, 2011a). All writing tasks have time constraints.

To illustrate the kind of errors encountered, the excerpt presented below was taken from a low scoring essay. In addition to spelling errors, it also involves multiple grammar errors and anomalous word order.

> the person who is going to be take a movie to saw the film is to takn to pass the star heroes movies . iam suppose to takn that is not valied is to distroy to take all heroneos

The corpus includes 3,000 essays, for a total of 963K words. The essays were selected equally from the two programs (4 tasks, 10 prompts per task, 75 essays per prompt), also covering full range of essay-scores (as a proxy for English proficiency levels) for each task. The majority of essays in this collection were written by examinees for whom English is not the first language. Out of the 1,500 TOEFL essays, 1,481 were written by non-native speakers of English (98.73%). Out of 1,500 GRE essays, 867 were written by non-native speakers of English (57.86%).

The annotation scheme for this project provides five classes of misspellings, as summarized in Table 1. The annotation effort focused specifically on misspellings, rather than on a wider category of orthographic errors in general. In annotation we deliberately ignored repeated words (e.g. *the the*), missing spaces (e.g. *...home.Tomorrow...*) and improper capitalization (e.g. *BAnk*). Many of the essays in our corpus have inconsistent capitalization. Some essays are written fully in capital letters. Although issues of proper capitalization fall under the general umbrella of orthographic errors, we do not consider them "spelling errors". In addition, in the annotated corpus, different spelling variants were acceptable. This consideration stems from the international nature of TOEFL and GRE exams – the examinees come from all around the world, being accustomed to either British, American, or some other English spelling standard; so, it is only fair to accept all of them.

Compilation and annotation of the corpus is a multi-stage project. At current stage, the exhaustive annotation effort focused on non-word misspellings. An in-house annotation software was developed for the project, as described by Flor and Futagi (2013). It automatically highlighted all non-words in a given text. The annotators were required to check all highlighted strings, and also scan the whole text for additional misspellings. They were encouraged to mark real-word misspellings as well (but that effort was not exhaustive). Classification of annotated strings was automatic. An annotated string was auto-marked as non-word if it was not found in the system dictionaries, and as a real-word misspelling if it was found in the system dictionaries. Annotators also marked multi-token errors, and the

annotation software automatically tagged them as "multi-token with non-word" (if at least one of the tokens was a non-word) or "multi-token real-words".

Type	Description	Count in corpus
1	single token non-word misspelling (e.g. "businees") also includes fusion errors (e.g. "niceday" for "nice day")	21,113 (87.04%)
2	misspelling (non-word token) for which no plausible correction was found	52 (0.21%)
3	multi-token non-word misspelling (e.g. "mor efun" for "more fun")	383 (1.58%)
4	single token real-word misspelling (e.g. "they" for "then")	2,284 (9.42%)
5	multi-token real-word misspelling (e.g. "with out" for "without")	425 (1.75%)
	Total	24,257

Table 1. *Classification of annotated misspellings in the ETS spelling corpus*

Each text was independently reviewed by two annotators, who are native English speakers experienced in linguistic annotation. Each misspelling was marked and the adequate correction was registered in annotation. A strict criterion was applied for agreement – two annotations had to cover exactly the same segment of text and provide same correction. Among all cases initially marked by annotators, they strictly agreed in 82.6% the cases. Inter-annotator agreement was then calculated over all words of the corpus. Agreement was 99.3%, Cohen's Kappa=0.85, $p<.001$. For all cases that were not in strict agreement, all differences and difficulties were resolved by a third annotator (adjudicator).

Overall, the annotated corpus of 3,000 essays has the following statistics. Average essay length is 321 words (the range is 28-798 words). 142 essays turned out to have no misspellings at all. Total spelling error counts are given in Table 1. Average error rate is 2.52% for all spelling errors, 2.2% for single-token non-words.

For each essay in this study, we obtained final essay scores as assigned by the TOEFL or GRE program. TOEFL essays are scored on a 1-5 scale. GRE essays are scored on a 1-6 scale. Using them as proxy "English proficiency scores", we divide the corpus into two subsets – essays of "higher quality" (HQ, score 4 and higher) and lesser quality (LQ, scores 1-3). Breakdown of counts for misspellings for two subsets is presented in Table 2. Proportion of misspellings (by token counts) is much higher among the LQ essays than among the HQ essays. Notably, both TOEFL and

GRE scoring guides do not require penalizing essays for spelling errors (ETS 2008, 2011a). In general, lower quality essays often involve many Spelling, Mechanics and Grammar errors, though their holistic scores also take into account their "narrative" and topical/argumentative quality (Ramineni *et al*., 2012a,b; Quinlan *et al*., 2009).

One additional aspect is error "severity", as indicated by the edit distance between the misspelled string and the correct form provided in the annotations. Table 3 provides such breakdown for single-token non-word misspellings in the corpus. Although the majority of misspellings are "minimal errors" (edit distance of 1), the amount of more severe errors is quite considerable. The lesser quality essays have a larger proportion of severe errors. To illustrate some of the more severe errors from this corpus: *foremore* (furthermore), *clacenging* (challenging), *QCCUPTION* (occupation, the error was originally in all caps), *naiberhouad* (neighborhood), *lungich* (language).

We did not attempt to classify the misspellings by classes of potential causes of errors. Traditional classifications consider typing errors vs. writer's ignorance of the correct spelling (including errors due to phonetic similarity). Since most of the essays in this corpus were written by non-native speakers of English, writer's knowledge was most probably involved. Two other factors may have contributed to proliferation of typing errors: the timed nature of the writing tasks and the fact that many examinees from around the world may not be sufficiently accustomed to a QWERTY keyboard.

Some researchers distinguish between "pure spelling errors" (e.g. typos) and "morphological errors" (e.g. *unpossible* when "impossible" was intended, or plural forms of words that do not have a marked plural – e.g. *knowledges*). In the annotated corpus, all such errors were marked as misspellings, without further sub-classification.

Essays	Higher Quality	Lesser Quality	All essays
Number of essays	1,342	1,658	3,000
Total word count	559,108	404,108	963,216
Misspellings (tokens)	6,829	14,284	21,113
Error rate	1.22%	3.53%	2.19%

Table 2. *Proportions of single-token non-word misspellings in ETS spelling corpus*

Edit distance	Higher Quality essays	Lesser Quality essays	All essays
1	5,812 (85.11%)	11,222 (78.56%)	17,034 (80.68%)
2	779 (11.41%)	2,120 (14.84%)	2,899 (13.73%)
3	158 (2.31%)	622 (4.35%)	780 (3.69%)
4+	80 (1.17%)	320 (2.24%)	400 (1.89%)
Total	6,829	14,284	21,113

Table 3. *Counts of single-token non-word errors by edit distance to correct form*

3. ConSpel system

The ConSpel system was designed and implemented as a fully automatic system for detection and correction of spelling errors. The current version is focused on single token non-word misspellings (including fusions). The system has two intended uses. One is to serve as a component in NLP systems for automatic evaluation of student essays. The other use is to facilitate automation for research on patterns of misspellings in student essays. This section describes the architecture and logic of the system.

3.1. Dictionaries and error detection

ConSpel has a rather simple policy for detection of non-word misspellings. A token in a text is potentially a misspelling if the string is not in the system dictionaries. Notably, a text may include some non-dictionary tokens that systematically should not be considered as misspellings. ConSpel has several parameterized options to handle such cases. By default, the system will ignore numbers, dates, Web and email addresses, and mixed alpha-numeric strings (e.g. “80MHz”).

ConSpel spelling dictionaries include about 360,000 entries. The core set includes 240,000 entries, providing a comprehensive coverage of modern English vocabulary. This lexicon includes all inflectional variants for a given word (e.g. “love”, “loved”, “loves”, “loving”), and international spelling variants (e.g. American and British English). Additional dictionaries include about 120,000 entries for international surnames and first names, and names for geographical places. Inclusion of person and place names is particularly important for an international setting, such as TOEFL and GRE examinations – essays written on these tests often include names of famous people and places from all over the world.

The use of a large lexicon of names drives down the rate of false alarms in error detection. However, it introduces a potential for misses – a misspelling may be undetected because such string is on the list of names. For example, "hince" is a misspelling of "hence", but "Hince" is also a common surname. In the past, inclusion of rare words and large lists of names was considered inadequate for spell-checkers, due to potential for misses in error-detection (Mitton, 1996). However, for modern NLP systems, in international settings, such as large scale language assessments, false alarms are detrimental, and high-coverage lexicons are essential. In principle, misspellings that happen to be identical to rare words or names, can be handled by real-word error detection (also known as "contextual spelling correction"). Similar issues arise in the domain of web query spelling correction (Whitelaw *et al.*, 2009). While this shifts the burden toward real-word error detection, it is better than having a high rate of false alarms.

Detection of errors is also influenced by text tokenization. The ConSpel tokenization subsystem tokenizes around punctuation, even if punctuation is "incorrect". For example, for a sequence like "*...travelled all nigt.They never...*", where a space is missing after the period, the system never presumes that *nigt.They* is a token – these are two tokens, and the first of them is misspelled. In the same vein, hyphenated forms (e.g. *semi-detached*) are also treated as multiple tokens. Abbreviations, such as *e.g.* are recognized as single tokens.

In the current version of ConSpel system, detection of non-word misspellings is implemented as a separate module – detection of errors is not related to any potential corrections. Thus the system can be used to flag errors without attempting to correct them. This also means that error detection rate is the same for whatever algorithms we choose to use for error correction (as presented further in this paper).

For evaluation against an annotated "gold standard" dataset, success rates are usually reported using measures of precision, recall and F1 score (Leacock *et al.*, 2010). Recall is defined as proportion of relevant materials retrieved, and precision is defined as proportion of retrieved materials that are true or relevant (Manning and Schütze, 1999). In this study, we use the standard definitions:

Recall: *# of annotated misspellings flagged by the system / # of all annotated misspellings*
Precision: *# of annotated misspellings flagged by the system / # of all tokens flagged by the system*
F1 score: *2 * precision * recall / (precision + recall)*

Table 4 presents ConSpel error detection rates for single-token non-word misspellings over 3,000 essays of the ETS Spelling Corpus. Recall is very high, but not perfect. The small number of misses were due to a) a few errors of tokenization, and b) the system ignored some "words with digits" – for example "10times" (annotated as misspelling in the corpus). Precision of error detection is also very high. ConSpel makes some false alarms, especially in the set of better-quality essays. False alarms are mostly due to names that are not yet in the dictionary, some

ad-hoc acronyms, and also due to some creative uses of language, like the term *Coca-Cola-ization* (reproduced here verbatim, ConSpel flagged *ization* as a misspelling, which it is not).

Essays	Corpus count of misspellings	Recall	Precision	F1
Whole corpus	21,113	99.84	98.79	99.31
Lesser quality	14,284	99.79	99.16	99.47
Higher quality	6,829	99.94	98.03	98.98

Table 4. *Error detection rates for single-token non-word misspellings in the ETS spelling corpus*

4. Correction of misspellings

For correction of single-token non-word misspellings, we use the following approach. For each detected misspelling, generate a set of candidate corrections, then rank the corrections – either in isolation, or by adding some contextual information.

The same dictionaries that are used for error detection are also the source of suggested corrections. Candidate suggestions for each detected misspelling are generated by returning all dictionary words that differ from the misspelling by a certain number of characters, up to a given threshold. We use the efficient Ternary Search Trie data structure (Bentley and Sedgewick, 1997) for candidate generation. The threshold is dynamic (depending on the length of a misspelled token), with a default value of 5. In addition to single-token candidates, ConSpel also generates multi-token candidates. This allows the system to correct fusion errors (e.g. “cando” when “can do” was intended). As noted by Mitton (2008), failures may occur at the candidate-generation stage – when the required word is not included among the initial set retrieved from the dictionary. Since ConSpel is intended to work on ELL data, and ELL misspellings can be quite dissimilar from the intended words, starting with a large number of candidates is a deliberate strategy to ensure that the adequate correction will be included in the candidate set. For each misspelled token, ConSpel typically generates more than a hundred correction candidates, and in some cases beyond a thousand candidates. Candidates are pruned during the re-ranking process, so that only a few candidates from the initial set survive to the final decision-making stage.

Candidate suggestions for each detected misspelling are ranked using a varied set of algorithms. ConSpel correction system is structured as a weighted ensemble of independent rankers. This allows researchers to switch certain algorithms on and off, so as to explore their effect and usefulness for the overall performance. For each

misspelling found in a text, each algorithm produces raw scores for each candidate. Then, scores for all candidates of a given misspelling are normalized into a 0-1 range, separately for each ranker. Finally, for each candidate, normalized scores are summed across rankers, using a set of constant weights. Thus, the score for a particular candidate correction is a linear combination of real-valued features:

$$\text{Total score for a Candidate Correction} = \sum w_A \cdot S_A$$

where w_A is the constant weight assigned to a particular algorithm (ranker, e.g. edit distance), and S_A is the score that that algorithm computed for the candidate, normalized vis-à-vis other candidate corrections of same misspelling. The coefficients (weights for rankers) used in this study were found experimentally (in the future we may use machine learning to optimize the process of assigning coefficients).

An error-model is often used to model the probability that certain words are typically misspelled in particular ways, e.g. "department" may be misspelled as "departmant" but rarely as "deparzment". Various approaches have been proposed to capture such regularities. An error model can be rule based, and can even be tuned to particular errors made by speakers of a specific L1 (Mitton and Okada, 2007). Other approaches (noisy channel models) include statistical character-confusion probabilities (Kernighan *et al.*, 1990; Tong and Evans, 1996) and substring confusion probabilities (Brill and Moore, 2000). To train an error model, a training set is needed consisting of string pairs – misspelling paired with the correct spelling of the word (Ristad and Yianilos, 1998). Whitelaw *et al.* (2009) demonstrated building an error model by leveraging Web services to automatically discover the misspelled/corrected word pairs. Given a misspelling to correct, a program can use such rules or probabilities to prefer certain corrections over other correction candidates. The ConSpel system does not use any error model. This was in part motivated by lack of resources to build a high-confidence error model, and in part due to our focus on the role of context in error-correction. Using context-informed re-ranking of candidate suggestions, without an error model, the ConSpel system accurately corrects spelling errors generated by non-native English writers, with almost the same rate of success as it does for writers who are native English speakers (Flor and Futagi, 2012).

4.1. Baseline

A set of algorithms in ConSpel perform "traditional" error correction – i.e. each misspelling is corrected in isolation, without considering the context. The backbone of the system is an edit distance module that computes orthographic similarity between each candidate and the original misspelling. Without an error model, simple unweighted edit distance is calculated (Levenstein, 1966; Damerau, 1964). Phonetic

similarity is calculated as edit distance between phonetic representation of the misspelling and phonetic representation of a candidate correction, which are produced using the Double Metaphone algorithm (Philips, 2000). Word frequency is computed for each candidate using a very large word-frequency data source (unigrams frequencies from Google Web1T corpus). For multi-token candidates, their *n*-gram frequency is retrieved from the *n*-gram database.

The direct way to evaluate automatic spelling correction is to consider how often the adequate target correction is ranked on top of all other candidates – a "Top1" evaluation. Another way is to consider how often the adequate target correction is found among the k-best candidate suggestions (Mitton, 2008; Brill and Moore, 2000). This allows to get an impression on how well a system approximates to the desired level of performance. We use k=5. In the following we report both "Top1" and "InTop5" evaluation results.

Table 5 presents evaluation results for error correction without context. The combination of orthographic edit distance, phonetic similarity and word frequency produces a very strong result – above 74% correct for Top1 evaluation. In 90% of the cases the adequate correction is among the top five corrections produced by the system. These results are used as baseline in evaluation of context-sensitive algorithms. Breakdown by essay quality reveals that the baseline algorithm performs better in HQ essays than in LQ essays ($p<.01$ for both Top1 and InTop5), which is expected given that LQ essays have more errors and more severe errors.

Algorithms involved	Top1			InTop5		
	Recall	Precision	F1	Recall	Precision	F1
Orthographic Similarity	54.45 50.47 62.78	53.96 50.25 61.61	54.20 50.36 62.19	84.98 82.62 89.91	84.22 82.27 88.24	84.60 82.44 89.07
Orthographic + Phonetic	64.28 61.02 71.09	63.70 60.76 69.78	63.99 60.89 70.43	87.29 85.46 91.13	86.51 85.10 89.44	86.90 85.28 90.28
Orthographic + Phonetic + Word Frequency	74.72 72.52 79.32	74.06 72.21 77.85	74.39 72.36 78.58	90.71 89.32 93.62	89.90 88.94 91.88	90.30 89.13 92.74

Table 5. *Evaluation results for spelling correction without context. In each cell: top – for whole corpus, middle – Low Quality essays, bottom – High Quality essays*

4.2. Word n*-grams*

Local context (several words around the misspelled word in the text) provides a lot of information for choosing the adequate correction. There is a long history of using word *n*-gram language models for spelling correction, for non-words – typically coupled with noisy-channel error-correction models (Brill and Moore, 2000, Zhang *et al.*, 2006; Cucerzan and Brill, 2004; Kernighan *et al.*, 1990), and for real-word errors (Wilcox-O'Hearn *et al.*, 2008; Mays *et al.*, 1991).

For a misspelling in text, we wish to choose a correction from a set of generated candidates. The misspelling occurs in a sequence of context tokens (see Figure 1), and words adjacent to the misspelling may help in choosing the adequate correction. Following Carlson & Fette (2007) and Bergsma *et al.* (2009), we want to utilize a variety of context segments, of different sizes and positions, that span the misspelled token. We look not only at the words that precede the misspelling in the text, but also at words that follow it. For a given misspelled token, there may be up to 2 bigrams that span it, up to 3 trigrams, etc. For each candidate correction, we check the frequency of its co-occurrence (in a language model) with the adjacent words in the text. With the advent of very large word *n*-gram language models, we can utilize large contexts (about 4 words on each side of a misspelling). Our current language model uses a filtered version of the Google Web1T collection, containing 1,881,244,352 word *n*-gram types of size 1-5, with punctuation included. Using the TrendStream tool (Flor, 2013), the language model is compressed into a database file of 11GB. During spelling correction, the same toolkit allows fast retrieval of word *n*-gram frequencies from the database.

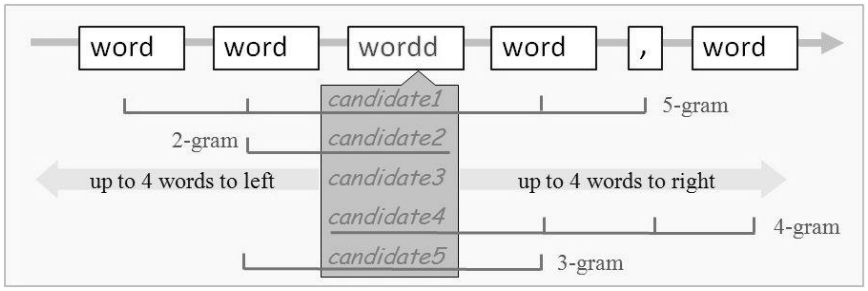

Figure 1. *Schematic illustration of capturing local context of a misspelling with overlapping n-grams*

Two particular details require more elaboration: how exactly are contexts defined and how we combine evidence (frequency counts) from *n*-grams of different sizes. Given a misspelling M in a context *xyzMefg* (here each letter stands symbolically for a word), for a candidate correction word C, the contextual word

bigrams are *zC* and *Ce*, trigrams are *yzC*, *zCe*, *Cef*, four-grams are *xyzC*, *yzCe*, etc. A competing correction candidate word D will have *n*-grams *zD*, *De*, *yzD*, *zDe*, *Def*, etc. How do we compare the total "support" for C with the total "support" for D, and other candidates? We have adopted the approach proposed by Bergsma *et al.* (2009), to sum the unweighted log-counts of all *n*-grams with a given candidate, even though those *n*-grams are of different sizes[1]. Thus, in the example above, the total evidence for candidate C (using just bigrams and trigrams) would be

log(count(zC))+log(count(Ce))+log(count(yzC))+log(count(zCe))+log(count(Cef))

and similarly when using higher order *n*-grams. In the end, for each candidate correction we get a single numeric value. Figure 2 presents a simplified schema of our algorithm. Using evidence from all around a misspelled word eliminates the need for language-model smoothing. If a candidate correction in combination with some context words results in an *n*-gram that is not found in the language model, then that combination gives no support to the candidate.

```
int position = Misspelling.getPositionInDocument();  //position of misspelled token in document
int maxwindow = Settings.getMaxWindowSize();  //max number of words on either side of misspelling
for (int window=2; int<=maxwindow; window++) {  //try all allowed window sizes
    //get local context:
   List<String> context = TextDocument.getLocalContext(position-window+1, position+window-1);
   for (Candidate candi : Misspelling.getListOfCandidateSuggestions()) {
       limit = Misspelling.putCandidateInContext(candi, context); //return limit for window slide
       for(int i=0; i<=limit i++) { //sliding window within local context
          //add to sum of contextual support:
          candi.ngramsSumLC += log(NgramsDatabase.getCount(context.subList(i, i+window));
       }     }     }
double topscore=Misspelling.getHighestLCscore();
for (Candidate candi : Misspelling.getListOfCandidateSuggestions()) {  //Normalization of scores
          candi.ngramsSumLC = candi.ngramsSumLC / topscore; }
```

Figure 2. *Algorithm (simplified Java pseudo-code) computing n-grams contextual support (by log counts) for correction candidates of one misspelling*

Table 6 presents evaluation results for a combination of baseline algorithms with the contextual algorithm that uses word *n*-gram frequencies (sum of log counts). For Top1 evaluation, bigrams provide about 6.35% improvement in the overall correction rate of the system, relative to the baseline. Adding trigrams raises the improvement to 8.74% above baseline. The added contribution of four-grams raises the improvement by another percent, to 9.86%. All differences from baseline are

1. There exists a different rationale, presented by Stehouwer and van Zaanen (2009): to sum evidence separately for each *n*-gram order and then sum across orders with some per-order-weights. We have tested both approaches and found that Bergsma *et al.* (2009) approach works better in our system. For another method, see Carlson & Fette (2007).

statistically significant, and each additional contribution is also statistically significant (p<.01). Adding five-grams raises improvement over baseline to 9.93%, but the addition is not statistically significant. The trend is quite clear. When utilizing *n*-gram frequencies as contextual support, bigrams provide the large improvement, and adding trigrams and four-grams significantly improves correction performance. Adding five-grams doesn't improve much. The results of InTop5 evaluation have a similar pattern. Substantial improvement over the strong baseline is achieved when extending to four-grams (each addition is statistically significant, p<.02). The small improvement with five-grams is not significant.

Breakdown of the results by essay quality (Table 7) reveals some interesting patterns (see also Figure 3.A). In both groups there is incremental improvement as high-order *n*-grams are added, and the LQ group consistently benefits more than the HQ group. All differences relative to respective baselines are statistically significant (p<.01). Top1 evaluation shows that in both groups adding trigrams provides significant improvement (p<.01). In the LQ group, adding four-grams significantly improves accuracy (more than 1%, p<.01). In the HQ group, adding four-grams is also beneficial (0.95%) and barely significant (p<.05). In both groups, adding five-grams provides very small improvement, which is not statistically significant. Continuous improvement of accuracy due to adding higher-order *n*-grams is also apparent in InTop5 evaluation. In the LQ group, adding trigrams over bigrams is significant (p<.01), adding four-grams is barely significant (p<.05) and adding five-grams is not significant. In the HQ group, only adding trigrams over bigrams is significant (p<.02).

	Top1			InTop5		
	Recall	Precision	F1	Recall	Precision	F1
Baseline	74.72	74.06	74.39	90.71	89.90	90.30
+*n*2	81.10 *6.38*	80.39 *6.33*	80.74 *6.35*	93.21 *2.50*	92.40 *2.50*	92.80 *2.50*
+*n*3	83.49 *8.77*	82.77 *8.71*	83.13 *8.74*	94.90 *4.19*	94.07 *4.17*	94.48 *4.18*
+*n*4	84.62 *9.90*	83.88 *9.82*	84.25 *9.86*	95.34 *4.63*	94.52 *4.62*	94.93 *4.63*
+*n*5	84.69 *9.97*	83.95 *9.89*	**84.32** *9.93*	95.61 *4.90*	94.78 *4.88*	95.19 *4.89*

Table 6. *Evaluation results for spelling correction that uses n-grams with log counts. Values in italics indicate improvement over the baseline. "+n2" means bigrams were used in addition to the baseline algorithms. "+n3" means bigrams and trigrams were used. "+n4" means four-grams were added, "+n5" means five-grams were added. All differences from baseline are significant (p<.01)*

	Top1			InTop5		
	Recall	Precision	F1	Recall	Precision	F1
Lesser Quality essays						
Baseline	72.52	72.21	72.36	89.32	88.94	89.13
+*n*2	79.62 *7.10*	79.28 *7.07*	79.45 *7.08*	92.99 *3.67*	92.59 *3.65*	92.79 *3.66*
+*n*3	81.76 *9.24*	81.40 *9.19*	81.58 *9.21*	94.25 *4.93*	93.84 *4.90*	94.04 *4.91*
+*n*4	82.95 *10.43*	82.59 *10.38*	82.77 *10.40*	94.72 *5.40*	94.31 *5.37*	94.51 *5.38*
+*n*5	83.05 *10.53*	82.69 *10.48*	**82.87** *10.50*	95.06 *5.74*	94.65 *5.71*	94.85 *5.72*
Higher Quality essays						
Baseline	79.32	77.85	78.58	93.62	91.88	92.74
+*n*2	84.81 *5.49*	83.31 *5.46*	84.05 *5.48*	95.52 *1.90*	93.83 *1.95*	94.67 *1.93*
+*n*3	87.13 *7.81*	85.59 *7.74*	86.35 *7.78*	96.25 *2.63*	94.55 *2.67*	95.39 *2.65*
+*n*4	88.09 *8.77*	86.54 *8.69*	87.31 *8.73*	96.65 *3.03*	94.94 *3.06*	95.79 *3.05*
+*n*5	88.11 *8.79*	86.55 *8.70*	**87.32** *8.74*	96.78 *3.16*	95.07 *3.19*	95.92 *3.18*

Table 7. *Evaluation results for spelling correction that uses n-gram frequencies, for LQ and HQ essays. Values in italics indicate improvement over the respective baseline*

4.3. Using n*-grams with PMI*

Using contextual *n*-gram frequencies has a long tradition for spelling correction. The logic is quite simple – a candidate correction that has higher co-occurrence with surrounding context is probably a better candidate. However, there can be a different, competing rationale. A better candidate might not be the one that occurs more often with context words, but one that has better affinity or "significance" of occurring in that context. This can be estimated by using statistical measures of association. There is an extensive literature on use of such measures for NLP, especially for detection/extraction of collocations (Pecina, 2009; Evert, 2008). After some experimentation, we found that normalized pointwise mutual information (Bouma, 2009) works rather well for spelling correction. It is defined as:

$$\text{Normalized PMI} = \left(\log_2 \frac{p(a,b)}{p(a)\,p(b)} \right) \Big/ -\log_2 p(a,b)$$

where *p(a,b)* is the observed probability of the sequence (a,b) in the corpus.[2] The formula extends to longer word sequences as well. Unlike the standard PMI (Manning and Schütze, 1999), normalized PMI (nPMI) has the property that its values are mostly constrained in the range {-1,1}, it is less influenced by rare extreme values, which is convenient for summing evidence from multiple *n*-grams. Additional experiments have shown that ignoring negative nPMI values works best (non-positive nPMI values are mapped to zero).[3] We define contextual *n*-grams in the same way as described in previous section, but for each *n*-gram, instead of using the *n*-gram frequency (log count), we calculate positive normalized PMI (PNPMI). For a given misspelling, for each candidate correction we sum PNPMI values from all relevant contextual *n*-grams.

Evaluation results are presented in Table 8. In Top1 evaluation of PNPMI, using just bigrams provides 7.8% improvement over the baseline (statistically significant, $p<.01$), towards an overall F1=82.2. Adding trigrams improves the performance by additional 2% (also statistically significant, $p<.01$). Adding four-grams adds 0.65% (this addition $p<.04$). Adding five-grams reverses the trend ($p<.03$). The InTop5 evaluation shows a similar trend – strong improvement over the baseline is achieved with bigrams, and adding trigrams and then four-grams provides fully 4.4% improvement over the baseline (each contribution is significant, $p<.01$). Addition of five-grams does not reverse the trend like in Top1, rather there is small improvement, but it is not statistically significant.

A breakdown by essay quality (Table 9) shows that the algorithm that uses *n*-grams with PNPMI is beneficial for each group: best improvement is 10.67% in LQ group, and 9.79% in the HQ group. Adding four-grams does not improve over trigrams in the LQ group, and only insignificant improvement in the HQ group. In both groups, adding five-grams reduces accuracy (although the reduction is not significant). For InTop5 evaluation, *n*-grams-with-PNPMI seem to have a stronger impact for LQ essays (up to 5.2% improvement over the strong 89% baseline) than for HQ essays (up to 3.3% improvement over the strong 92% baseline). In both cases, the differences are significant, $p<.01$. Here, improvement continues even with addition of five-grams – in both groups statistically significant ($p<.02$) relative to trigrams, but not significant relative to four-grams. The overall trend of the improvement is different in Top1 and InTop5 evaluations. With Top1, improvement in LQ group peaks with three-grams, an in the HQ group it peaks with four-grams. In the InTop5 evaluation, improvement keeps rising even when five-grams are added, in both groups. Using long *n*-grams helps promoting the adequate candidates into the top five.

---

2. The corpus is our filtered version of the Google Web1T, the same one mentioned in the previous section. The database computes unigram probabilities, joint probabilities p(a,b), etc., on the fly, based on stored count values. For technical details see Flor (2013).

3. This kind of practice is described by Bullinaria and Levy (2007), also Mohammad and Hirst (2006), and was first suggested by Church and Hanks (1990).

Figure 3 plots F-scores of Top1 evaluation for spelling correction with *n*-grams, comparing the use of log counts and PNPMI. Overall, the results for both measures are similar. Both measures provide a very strong improvement over the baseline. For both measures the improvement rises when longer *n*-grams are added, up to four-grams. Both approaches seems to level at that point (the slight degradation with PNPMI five-grams is not significant). One pattern to note is that results achieved with PNPMI seem to be slightly better than those achieved with log counts. It is particularly pronounced for the set of high quality essays. In the HQ set, PNPMI achieves F1=86.15 with bigrams, 88.22 when trigrams are added, and 88.37 when four-grams are added. Log counts achieve 84.05, 86.35 and 87.31 respectively (the respective differences are statistically significant with p<.03). The differences between PNPMI and log counts are not significant in the set of lower quality essays.

	Top1			InTop5		
	Recall	Precision	F1	Recall	Precision	F1
Baseline	74.72	74.06	74.39	90.71	89.90	90.30
+*n*2	82.57 *7.85*	81.85 *7.79*	82.21 *7.82*	93.21 *2.50*	92.40 *2.50*	92.81 *2.50*
+*n*3	84.49 *9.77*	83.75 *9.69*	84.12 *9.73*	93.95 *3.24*	93.13 *3.23*	93.54 *3.23*
+*n*4	**85.14** *10.42*	**84.40** *10.34*	**84.77** *10.38*	95.14 *4.43*	94.31 *4.41*	94.72 *4.42*
+*n*5	84.47 *9.75*	83.74 *9.68*	84.10 *9.71*	95.30 *4.59*	94.47 *4.57*	94.88 *4.58*

Table 8. *Evaluation results for spelling correction with n-grams using positive normalized PMI. Values in italics indicate improvement over the baseline. All differences from baseline are significant (p<.01)*

	Top1			InTop5		
	Recall	Precision	F1	Recall	Precision	F1
Lesser Quality essays						
Baseline	72.52	72.21	72.36	89.32	88.94	89.13
+*n*2	80.48 *7.96*	80.13 *7.92*	80.30 *7.94*	92.28 *2.96*	91.88 *2.94*	92.08 *2.95*
+*n*3	83.23 *10.71*	82.87 *10.66*	**83.05** *10.68*	93.77 *4.45*	93.36 *4.42*	93.56 *4.43*
+*n*4	83.22 *10.70*	82.86 *10.65*	**83.04** *10.67*	94.40 *5.08*	93.99 *5.05*	94.19 *5.06*
+*n*5	82.53 *10.01*	82.18 *9.97*	82.35 *9.99*	94.54 *5.22*	94.13 *5.19*	94.33 *5.20*
Higher Quality essays						
Baseline	79.32	77.85	78.58	93.62	91.88	92.74
+*n*2	86.92 *7.60*	85.39 *7.54*	86.15 *7.57*	95.17 *1.55*	93.48 *1.60*	94.32 *1.58*
+*n*3	89.02 *9.70*	87.44 *9.59*	88.22 *9.64*	96.22 *2.60*	94.52 *2.64*	95.36 *2.62*
+*n*4	89.16 *9.84*	87.59 *9.74*	**88.37** *9.79*	96.69 *3.07*	94.98 *3.10*	95.83 *3.09*
+*n*5	88.52 *9.20*	86.95 *9.10*	87.73 *9.15*	96.90 *3.28*	95.18 *3.30*	96.03 *3.29*

Table 9. *Evaluation results for spelling correction with n-grams and PNPMI, for LQ and HQ essays. Values in italics indicate improvement over respective baseline. All differences from baseline are significant (p<.01)*

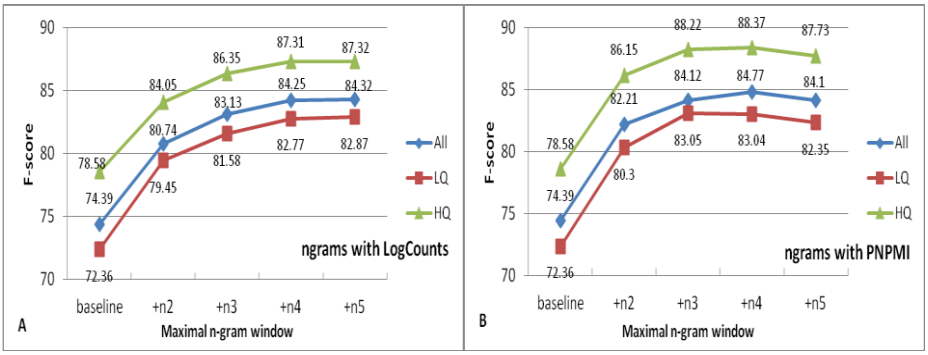

Figure 3. *F-score values from evaluation results (Top1) of spelling correction with n-grams, (A) using log-counts, (B) using PNPMI. LQ: lower quality essays, HQ: higher quality essays, All: all essays*

4.3. Semantic relatedness for spelling correction

Semantic relatedness between words, as exhibited in the lexical content of a text, can be useful for spelling correction. Consider a misspelling and some candidate suggestions. A candidate may be considered more plausible (i.e. ranked higher than other candidates) if it is semantically related to some other words in the text that is being corrected. Intuitively, a good correction candidate should "fit" into the semantic fabric of the text, and the better the "fit" is, the higher ranking it should achieve. Basically, we can take each candidate and measure how well it fits with each word in the text (in the vicinity of the misspelling).

The idea of checking "semantic fit" was proposed by Budanitsky and Hirst (2006) for detecting and correcting real-word misspellings. They used WordNet for measuring semantic relatedness. To the best of our knowledge, there is no prior use of "semantic relatedness" for improving correction of non-word misspellings.

The current proposal requires two components – a measure of fit and a resource that can provide pairwise estimations of semantic relatedness. For example, if the misspelling is *carh* and one of the candidate corrections is *car*, this correction may look more plausible if the text around the misspelling includes such words as *engine*, *automobile* and *roads*. On the other hand, a correction candidate *card* may gain plausibility if the surrounding words include *aces* and *casino*, or *payment* and *expired*. Note that such semantic fit is not restricted to semantically similar words (e.g. *car-automobile*), but can benefit from any semantic relatedness (e.g. *card-payment*), even without knowing what exactly the relation is or how to label it. WordNet is often used as a resource for obtaining estimates for semantic relations and similarity (Zhang, Gentile and Ciravegna, 2012). However, WordNet lacks enough coverage. For example it is quite difficult to obtain from WordNet such thematic relations as *dog-bark*, *card-expire*, or *city-traffic*.

To attain a wide-coverage resource, we use a first-order co-occurrence word-space model (Turney and Pantel, 2010; Baroni and Lenci, 2010). The model was generated from a corpus of texts of about 2.5 billion words[4], counting co-occurrences in paragraphs. The (largely sparse) matrix of 2.1x2.1 million word types and their co-occurrence frequencies, as well as single-word frequencies, is efficiently compressed using the TrendStream tool (Flor, 2013), resulting in database file of 4.7GB. During spelling correction, the same toolkit allows fast retrieval of word probabilities and statistical associations for pairs of words.[5]

With a wide-coverage word-association resource, there are two more parameters to consider. We need to consider which statistical association measure to use. As

4. About 2 billion words come from the Gigaword corpus (Graf and Cieri, 2003), which is a news corpus. Additional 500 million words come from an internal corpus at ETS, with texts from popular science and fiction genres.

5. The database storing the distributional word-space model includes counts for single words and for word pairs. Association measures are computed on the fly.

with the *n*-grams approach, we have found that normalized PMI (Bouma, 2009) works rather well. Note that unlike *n*-grams, where we consider the probabilities or strength of whole sequences, here we consider pair-wise strengths: for each correction candidate (of a given misspelling), we want to sum the pair-wise strengths of its association with "every" word in the context. One important observation is that even a good candidate need not be strongly related to every word in its context – even in a cohesive text, a word is typically strongly related only to some of the neighboring words (Hoey 2005, 1991). With a measure like nPMI, some pairwise associations have negative values. Here again, we have found that it is beneficial to disregard negative association values, and sum only the positive evidence. Our choice measure is positive normalized PMI (PNPMI). In addition we disregard context words that belong to a stoplist (e.g. determiners and common prepositions). The schematic version of the algorithm is presented in Figure 4.

Evaluation results on the corpus data are presented in Table 10. Semantic relatedness provides 3.8% improvement in the overall correction rate of the system, relative to the baseline. The improvement is slightly better for lower quality essays (3.9%) than for higher quality texts (3.6%). All improvements are statistically significant ($p<.01$). InTop5 evaluation shows that semantic relatedness also has about 1% contribution ($p<.01$) for promoting candidates into the top five, relative to a strong baseline, as we have with this corpus.

Another consideration was how far to look around the misspelling – the possibilities ranging from the whole text to just the few neighboring words. We have experimented with a context window of *k* words to the left and to the right of the misspelled word in the text, with *k* values of 5 to 40 (in increments of 5), thus using neighborhoods of 10, 20, 30 to 80 words. The results were very similar to those presented in Table 10 and the tiny differences were not statistically significant. The F-scores from these runs are plotted in Figure 5. The improvement over the baseline is clearly visible. As for context size, it seems the valuable information is contained within the window of ±5 words around the misspelling, expanding the context beyond that does not help (but does not harm either).

```
int position = Misspelling.getPositionInDocument();  //ordinal position of misspelled token in document
int quota = Settings.getQuota();  //max num. of words to search on either side of misspelling.
List<String> contextualList = TextDocument.generateContextList(position,quota);
//For each candidate, sum evidence from all contextual words:
for (Candidate candi : Misspelling.getListOfCandidateSuggestions()) {
   for(String contextWord : contextualList) {
      candi.SRsupport += WordAssociationsDatabase.getPNPMI(candi.word, contextWord);     } }
double topscore=Misspelling.getHighestSRscore();
for (Candidate candi : Misspelling.getListOfCandidateSuggestions()) {  //Normalization of scores
        candi.SRsupport = candi.SRsupport / topscore; }
```

Figure 4. *Algorithm (Java pseudo-code) computing Semantic Relatedness contextual support for correction candidates of one misspelling*

	Top1			InTop5		
	Recall	Precision	F1	Recall	Precision	F1
All essays						
Baseline	74.72	74.06	74.39	90.71	89.90	90.30
+SR	78.54 *3.82*	77.86 *3.80*	78.20 *3.81*	91.73 *1.02*	90.93 *1.03*	91.33 *1.03*
Lesser Quality essays						
Baseline	72.52	72.21	72.36	89.32	88.94	89.13
+SR	76.45 *3.93*	76.12 *3.91*	76.28 *3.92*	90.49 *1.17*	90.10 *1.16*	90.29 *1.16*
Higher Quality essays						
Baseline	79.32	77.85	78.58	93.62	91.88	92.74
+SR	82.91 *3.59*	81.44 *3.59*	82.17 *3.59*	94.32 *0.70*	92.65 *0.77*	93.48 *0.74*

Table 10. *Evaluation results for spelling correction with Semantic Relatedness, using context size of up to 20 words (10 on each side of a misspelling). Values in italics indicate improvement over the baseline. All differences from respective baselines are significant (p<.01)*

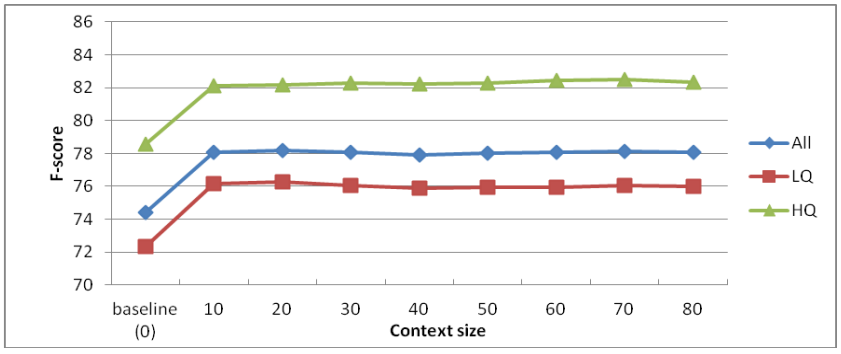

Figure 5. *F-score values from evaluation results of spelling correction with Semantic Relatedness (word associations), using different sizes of context*

4.5. Déjà vu – taking advantage of recurring words

Here we present another context-sensitive algorithm that utilizes non-local context in an essay. Content words have some tendency of recurrence in same text (Halliday and Hasan, 1976; Hoey, 1991). This tendency of lexical repetition is utilized, for example, for computing a type/token ratio (Yu, 2010). Lexical repetition

is also the backbone for constructing lexical chains (Morris and Hirst, 1991). In student essays, it often happens that the same word (type) is used several times in a text, and that some of those tokens are misspelled. This can have a direct implication for re-ranking of correction candidates. Given a misspelled token in a text and a set of correction-candidates for that token, for each candidate we check whether same word (or inflectional variant) occurs elsewhere in the text. Consider this example:

… We wanted to look at the stars. … the stard was …

Misspelling *stard* may have several candidate corrections, including *star*, *start* and *stand* (for illustration we list those having same edit distance, but it is not a prerequisite). The presence of the word *stars* in the text may be used to promote *star* in the ranking of candidates in this case. It might be wrong, and the adequate correction in this case could be *stand*, but in general this "Dejavu" approach can be useful.

Given a candidate correction for a specific misspelled token in a text, the Dejavu Algorithm looks over the whole essay. When same word type, or its inflectional variant, is encountered, the candidate is strengthened. The amount of strengthening is inversely proportional to distance between the encountered word and the misspelled token. This stems from a conjecture that a repetition close to the site of misspelling may be more relevant (for that misspelling) than a distant repetition (say a hundred words away). We use 1/sqrt(1+distance) as the score that a candidate gets when encountering each related instance in the text (all such scores for a given candidate are summed; summed scores for all candidates of a given misspelling are normalized). Schematic version of the algorithm is presented in Figure 6.

The idea of utilizing repeated words is somewhat similar to the notion of cache-based language-model adaptation (Kuhn and De Mori, 1990), which was proposed in the domain of speech recognition. However, our current approach is different – we do not use a statistical language model in this case, and we extend lexical coverage with inclusion of inflectional variants.

An additional variant of Dejavu Algorithm is useful for treating systematic misspellings. Those may occur due to mistyping, but it also often happens when a writer does not know how to properly spell a given word in English, and so all (or most) intended instances of that word in an essay are misspelled. For example:

the responsibility of the ex-finance mininster to argue ,but he will not respond ,but the other member will argue ,asif the mininster ...

In our corpus, there are even essays where all instances of the same word are misspelled in different ways. For example, in one essay the word "knowledge" was used four times, misspelled as *knowlege*, *knowleges* and *knowledges* (twice).

The advanced handling is as follows. For a given misspelled token, for each candidate correction, the Dejavu Algorithm searches not only in the essay text (for

"repetitions"), but also looks into the lists of candidate corrections of other misspelled tokens in the text. If a corresponding word (or its inflectional variant) is found in such list, our candidate is strengthened with a score of S_{CC}/sqrt(1+distance), where S_{CC} is current normalized overall strength of the "corresponding candidate" in the other list.[6] Thus, if a word is systematically misspelled in a document, Dejavu will considerably strengthen a candidate correction that appears as a (good) candidate for multiple misspelled tokens – a kind of mutual optimization across the whole text. The advanced part of the algorithm is also presented in Figure 6.

Evaluation results for Dejavu Algorithm on the corpus data are presented in Table 11. In Top1 evaluation, the algorithm improves the correction rate by 3.8% ($p<.01$) relative to the baseline. The results are similar in the breakdown by essay quality: about 3.89% correction improvement for LQ essays, and above 3.73% correction improvement for HQ essays (in both cases $p<.01$). The InTop5 evaluation shows a small improvement over the baseline, 1% ($p<.01$). The improvement is larger in the LQ group (1.1%, $p<.01$) and very small in the HQ group (0.7%, $p<.04$).

	Top1			InTop5		
	Recall	Precision	F1	Recall	Precision	F1
All essays						
Baseline	74.72	74.06	74.39	90.71	89.90	90.30
+Dejavu	78.58 *3.86*	77.88 *3.82*	78.23 *3.84*	91.71 *1.00*	90.89 *0.99*	91.30 *0.99*
Lesser Quality ssays						
Baseline	72.52	72.21	72.36	89.32	88.94	89.13
+Dejavu	76.42 *3.90*	76.10 *3.89*	76.26 *3.89*	90.44 *1.12*	90.06 *1.12*	90.25 *1.12*
Higher Quality essays						
Baseline	79.32	77.85	78.58	93.62	91.88	92.74
+Dejavu	83.09 *3.77*	81.55 *3.70*	82.31 *3.73*	94.36 *0.74*	92.61 *0.73*	93.48 *0.73*

Table 11. *Evaluation results for spelling correction with Dejavu algorithm. Values in italics indicate improvement over the baseline*

6. The better the rank of the candidate is in other lists, the stronger the candidate gets for the current misspelling. This presumes that candidates are already ranked and sorted by the other algorithms. There is no issue of infinite loops – the advanced part of the Dejavu algorithm runs after all other ranking algorithms – it uses the other rankings for its calculations.

```
int position = Misspelling.getPositionInDocument();//ordinal position of misspelled token in document
double val;
//For each candidate, find matching words in the text and sum evidence (Basic Dejavu):
//loop on all candidate corrections of the Misspelling:
for (Candidate candi : Misspelling.getListOfCandidateSuggestions()) {
   for(Token tokenword : TexDocument.getAllWordTokens()) { //loop on all words in document
      if (tokenword.position == position) continue;
      if (tokenword.isSameWord(candi.word)) { val = Settings.getDejavuScoreForIdentical(); }
      else if (tokenword.isInflectionalVatiant(candi.word)) { val=Settings.getDejavuScoreForInflectional();}
      else { val=0.0; }
      val = val / sqrt( 1+ abs(tokenword.position - position)) //weight by distance
      candi.DejavuSupport += val; //add to sum of support values
   }  }
//For each candidate, peek into the candidate lists of other misspelled tokens (Advanced Dejavu):
for(MisspellingObject MO : TexDocument.getAllMisspellings()) { //loop on all Misspellings in document
   if (Misspelling==MO) continue;
   //loop on all candidate corrections of the Misspelling
   for (Candidate candi : Misspelling.getListOfCandidateSuggestions()) {
      //loop on all candidates in other misspellings
      for(Candidate candiElsewhere : MO.getListOfCandidateSuggestions()) {
         if ( isSameWord(candi.word, candiElsewhere.word)) {val=Settings.getDejavuScoreForIdentical(); }
         else if ( isInflectionalVatiant(candi.word, candiElsewhere)) {
               val=Settings.getDejavuScoreForInflectional(); }
          else { val=0.0; }
          //weight by distance:
          val = val * candiElsewhere.getTotalScore / sqrt( 1+ abs(position - MO.getPositionInDocument()));
          candi.DejavuSupport += val; //add to sum of support values
      }   }  }
double topscore=Misspelling.getHighestDejavuScore();
for (Candidate candi : Misspelling.getListOfCandidateSuggestions()) {  //Normalization of scores
    candi.DejavuSupport = candi.DejavuSupport / topscore; }
```

Figure 6. *Algorithm (Java pseudo-code) computing Dejavu contextual support for correction candidates of one misspelling*

5. Biased correction

A different approach to the notion of "context" is the idea of biasing error-correction to the particular topic of the text. If the topic of a text is known, a list of topic-specific words may be given preferential status when correcting misspellings for that text. Strohmaier *et al.* (2003) demonstrated post-correction of OCR output, with topic-specific dictionaries. In that study, dictionaries where automatically generated from the vocabulary of Web pages from given topical domains, but selection of topics was manual. Wick, Ross, and Learned-Miller (2007) described the use of dynamic topic models to post-correct (simulated) OCR output. In that study, topic models were learned from collections of newsgroup documents. Some of held-out documents were artificially "corrupted" and used as correction-test cases. In both studies, topical words were given extra weight in candidate ranking, leading to improved overall correction rate.

In current work, we use a weak notion of topical bias. Essays written to TOEFL and GRE prompts are prompt-specific. The prompts are open-ended and only weakly-constrained, and the essays may exhibit considerable variability, some are even off-topic. However, the text of the prompt itself is a very strong anchor for each essay. Thus, we use the content words of a prompt (and their inflectional variants), as a biasing list for error correction of essays written to that prompt. For example, suppose there is a text with misspelling *pawer* and one of the candidate corrections is *power*. If *power* (or *powers*) appeared in the corresponding prompt, then the candidate may be strengthened in the ranking of candidates for *pawer*. Schematic version of the algorithm is presented in Figure 7.

Evaluation results for topical biasing algorithm, on the corpus data, are presented in Table 12. The contribution of biased spelling correction was negligible and never statistically significant. One possible reason for this result may be that many of the 40 prompts included in this study were quite short, generic and open-ended. The following prompt (not in the corpus, from ETS 2011a) illustrates this:

> As people rely more and more on technology to solve problems, the ability of humans to think for themselves will surely deteriorate. [Write a response in which you discuss the extent to which you agree or disagree with the statement and explain your reasoning for the position you take. In developing and supporting your position, you should consider ways in which the statement might or might not hold true and explain how these considerations shape your position.]

However, there is one particular kind of task where prompts are rich and include plenty of "context words". The TOEFL Integrated task (ETS, 2011b) is posed as follows. The examinee is presented with a reading passage that presents some arguments on a given topic. After reading the passage, the examinee listens to an audio recording, where a narrator presents some contrary arguments on the same topic. The task is to summarize and relate both opinions. The ETS Spelling Corpus contains 750 essays written to 10 prompts from the TOEFL Integrated task. For each of these prompts, "prompt vocabulary" was obtained, including words from the reading passage and the audio lecture. For these prompts, the bias lists include about 240 unique content words (types) per prompt, whereas for the other prompts in the corpus, bias lists have fewer than 70 unique content words (types) per prompt. Moreover, due to the nature of the TOEFL Integrated task (essentially a summarization/retelling task), it seems more plausible that words from the prompt will be reused in a response essay.

Performance of the biasing correction algorithm was evaluated separately on the 750 essays from the TOEFL Integrated task. Results are given in Table 13. Here, the contribution of biased spelling correction becomes evident. The improvement over baseline was 5.3% for Top1, and 2.4% for InTop5 evaluation (which has a very strong baseline). Breakdown by essay quality confirms that biased correction

improves over baseline, for both LQ and HQ essays. In Top1 evaluation, the improvement was greater in the HQ group (7.16%) than in the LQ group (4.47%). A similar trend is seen in the InTop5 evaluation.

```
double val;
//For each candidate, find matching words in the biasing dictionary and sum evidence:
//loop on all candidate corrections of the Misspelling:
for (Candidate candi : Misspelling.getListOfCandidateSuggestions()) {
   for(Word w : BiasingDictionary.getAllWords ()) { //loop on all words in biasing dictionary
      if ( isSameWord(candi.word, w)) { val = Settings.getBiasScoreForIdentical(); }  //default 1.0
      else if ( isInflectionalVariant(candi.word, w)) { val = Settings.getBiasScoreForInflectional(); }  //0.8
      else { val=0.0; }
      candi.BiasSupport += val; //add to sum of support values
   }    }
double topscore=Misspelling.getHighestBiasScore(); // highest BiasSupport among all candidates
for (Candidate candi : Misspelling.getListOfCandidateSuggestions()){  //Normalization of scores
    candi. BiasSupport = candi.BiasSupport / topscore; }
```

Figure 7. *Algorithm (Java pseudo-code) computing Biasing contextual support for correction candidates of one misspelling*

	Top1			InTop5		
	Recall	Precision	F1	Recall	Precision	F1
All essays						
Baseline	74.72	74.06	74.39	90.71	89.90	90.30
+Bias	75.22 *0.50*	74.55 *0.49*	74.88 *0.49*	90.93 *0.22*	90.12 *0.22*	90.52 *0.22*
Lesser Quality essays						
Baseline	72.52	72.21	72.36	89.32	88.94	89.13
+Bias	72.95 *0.43*	72.65 *0.44*	72.80 *0.44*	89.50 *0.18*	89.11 *0.17*	89.30 *0.17*
Higher Quality essays						
Baseline	79.32	77.85	78.58	93.62	91.88	92.74
+Bias	79.87 *0.55*	78.40 *0.55*	79.13 *0.55*	93.94 *0.32*	92.21 *0.33*	93.07 *0.33*

Table 12. *Evaluation results for biased spelling correction. Values in italics indicate improvement over the baseline; none are statistically significant*

	Top1			InTop5		
	Recall	Precision	F1	Recall	Precision	F1
All essays (750)						
Baseline	73.91	73.95	73.93	90.14	90.18	90.16
+Bias	79.29 *5.38*	79.22 *5.27*	79.26 *5.33*	92.62 *2.48*	92.53 *2.35*	92.57 *2.41*
Lesser Quality essays (452)						
Baseline	73.12	73.22	73.17	89.42	89.55	89.48
+Bias	77.59 *4.47*	77.70 *4.48*	77.64 *4.47*	91.63 *2.21*	91.75 *2.20*	91.69 *2.21*
Higher Quality essays (298)						
Baseline	75.49	75.38	75.43	91.56	91.44	91.50
+Bias	82.65 *7.16*	82.54 *7.16*	82.59 *7.16*	94.57 *3.01*	94.45 *3.01*	94.51 *3.01*

Table 13. *Evaluation results for biased spelling correction, with "rich-prompt" subset of the spelling corpus. Values in italics indicate improvement over the baseline. All improvements over baseline are statistically significant (p<.01)*

6. Combining contextual algorithms

We have also experimented with combining the various contextual algorithms: *n*-grams summing log-counts, *n*-grams summing PNPMI, semantic relatedness, word-repetitions (Dejavu) and biasing. All combinations are considered as additions to the baseline algorithms.

Evaluation results are presented in Table 14 and in Figures 8 and 9. The combination of Semantic Relatedness and Bias is counterproductive (decreases F1 score below what is achieved with Semantic Relatedness, p<.01). The combinations of Dejavu and Bias and that of Dejavu and Semantic Relatedness are also counterproductive, although the respective differences from using just Dejavu are not statistically significant. Algorithms that use *n*-grams are clearly the most effective. Adding Bias to *n*-gram algorithms has no effect. Adding Semantic Relatedness to either of the *n*-grams algorithms provides very little improvement, not statistically significant. Adding Dejavu to *n*-grams-with-log-counts is counterproductive (p<.01). However, adding Dejavu to *n*-grams-with-PNPMI is effective, raising F-score from 84.77 to 85.36 (p<.05). Combination of the two *n*-grams-based algorithms (F1=85.72) turns out to be more effective than either of them alone (84.32 and 84.77), and the added improvement is statistically significant (p<.01 in both cases). The best result is achieved by combining the two *n*-gram-based algorithms and Semantic Relatedness (F1=85.87), although the addition is not statistically significant.

	Top1			InTop5		
	Recall	Precision	F1	Recall	Precision	F1
Baseline	74.72	74.06	74.39	90.71	89.90	90.30
Baseline with one contextual algorithm						
LC	84.69 *9.97*	83.95 *9.89*	84.32 *9.93*	95.61 *4.90*	94.78 *4.88*	95.19 *4.89*
PNPMI	85.14 *10.42*	84.40 *10.34*	84.77 *10.38*	95.14 *4.43*	94.31 *4.41*	94.72 *4.42*
SR	78.54 *3.82*	77.86 *3.80*	78.20 *3.81*	91.73 *1.02*	90.93 *1.03*	91.33 *1.03*
Dejavu	78.58 *3.86*	77.88 *3.82*	78.23 *3.84*	91.71 *1.00*	90.89 *0.99*	91.30 *0.99*
Bias	75.22 *0.50*	74.55 *0.49*	74.88 *0.49*	90.93 *0.22*	90.12 *0.22*	90.52 *0.22*
Baseline with combination of contextual algorithms						
SR & Bias	77.44 *2.72*	76.75 *2.69*	77.09 *2.70*	91.96 *1.25*	91.14 *1.24*	91.55 *1.24*
Dejavu & Bias	77.90 *3.18*	77.20 *3.14*	77.55 *3.16*	91.78 *1.07*	90.97 *1.07*	91.37 *1.07*
SR & Dejavu	78.31 *3.59*	77.61 *3.55*	77.96 *3.57*	92.30 *1.59*	91.48 *1.58*	91.89 *1.58*
LC & Dejavu	83.75 *9.03*	83.03 *8.97*	83.39 *9.00*	95.28 *4.57*	94.45 *4.55*	94.86 *4.56*
LC & Bias	84.20 *9.48*	83.48 *9.42*	83.84 *9.45*	95.48 *4.77*	94.65 *4.75*	95.06 *4.76*
PNPMI & Bias	84.62 *9.90*	83.89 *9.83*	84.25 *9.86*	95.42 *4.71*	94.59 *4.69*	95.00 *4.70*
LC &SR	84.80 *10.08*	84.06 *10.00*	84.43 *10.04*	95.63 *4.92*	94.80 *4.90*	95.21 *4.91*
PNPMI & SR	85.25 *10.53*	84.51 *10.45*	84.88 *10.49*	95.34 *4.63*	94.51 *4.61*	94.92 *4.62*
PNPMI & Dejavu	85.74 *11.02*	84.99 *10.93*	85.36 *10.97*	95.34 *4.63*	94.52 *4.62*	94.93 *4.63*
PNPMI & LCL	86.09 *11.37*	85.35 *11.29*	85.72 *11.33*	95.62 *4.91*	94.79 *4.89*	95.20 *4.90*
PNPMI & LC & SR	**86.25** *11.53*	**85.50** *11.44*	**85.87** *11.48*	**95.69** *4.98*	**94.85** *4.95*	**95.27** *4.96*

Table 14. *Evaluation results for combined contextual methods, whole corpus. Values in italics indicate improvement over the baseline. All improvements over baseline are statistically significant (p<.01). LC: n-grams with log counts, PNPMI: n-grams with PNPMI, SR: Semantic Relatedness*

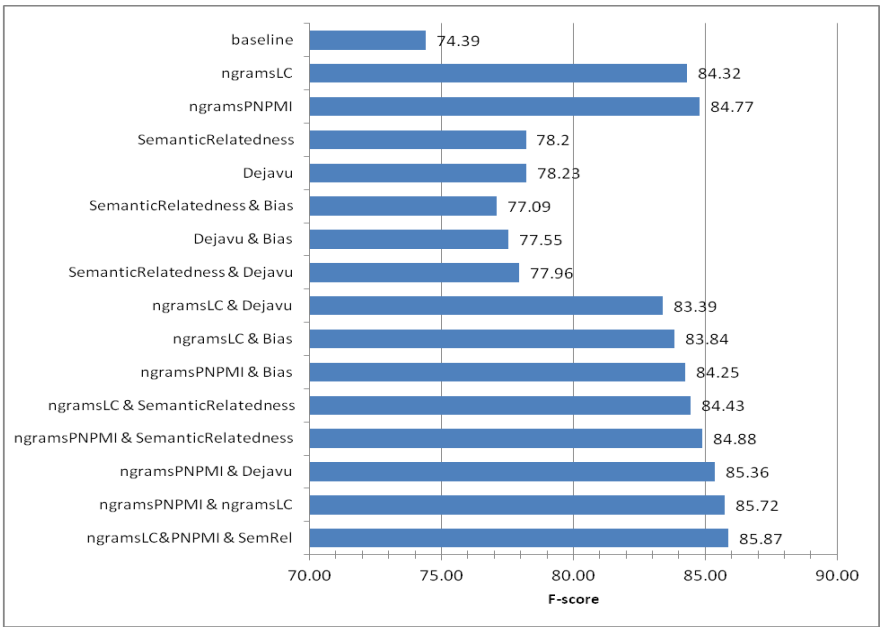

Figure 8. *F-score values from evaluation results (Top1) of spelling correction with various combinations of contextual algorithms, over the full corpus of essays*

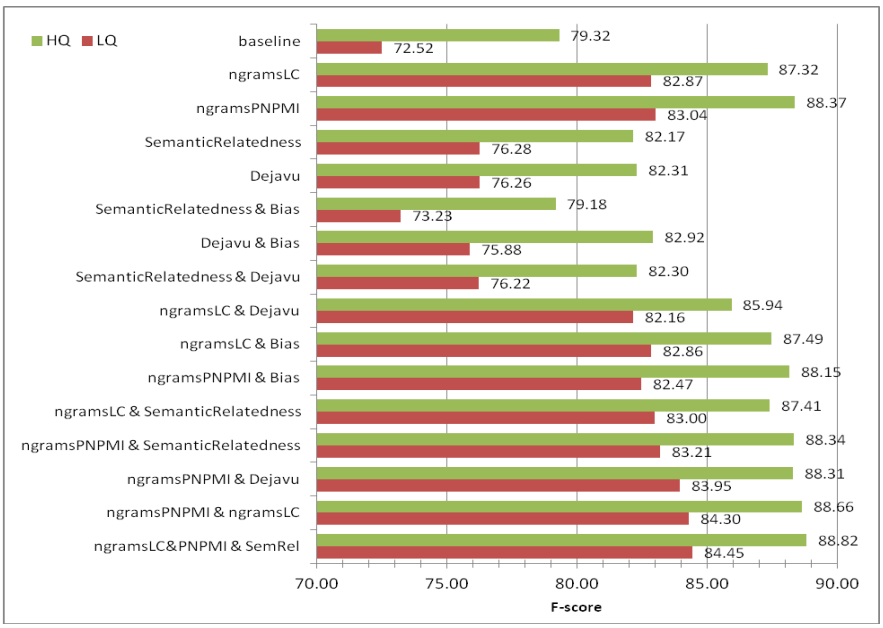

Figure 9. *F-score values from evaluation results (Top1) of spelling correction with various combinations of contextual algorithms LQ/HQ: lower/higher quality essays*

Evaluation with breakdown by essay quality (Figure 9) shows similar results. Only the combination of two *n*-grams-based algorithms turns out to be more effective than either of them alone, in both lower and higher quality sets, and the added improvement is statistically significant ($p<.01$ in both cases as compared to *n*-grams-log-counts algorithm alone). When this combination is compared to *n*-grams-PMPMI, the improvement is significant ($p<.01$) in the HQ set, but not significant in the LQ set. The best result is each set is achieved by combining the two *n*-gram-based algorithms and Semantic Relatedness, but in both sets the addition of Semantic Relatedness is not statistically significant over the combination of the *n*-grams-based algorithms.

7. Discussion

In a previous study (Flor and Futagi, 2012), we conducted a comparison between ConSpel and two widely used spellchecking systems – Aspell (Atkinson, 2011) and the speller from MS Office (MS Word) 2007. The ConSpel system showed better detection of single-token misspellings than MS Word or Aspell, even when those system were given the ConSpel dictionary. In that study, ConSpel with contextual algorithms also outperformed MS Word and Aspell in automatic spelling correction. For spelling correction, ConSpel had F1=77.93, while MS Word (with ConSpel dictionary) had F1=70.56, and Aspell had F1=51.83. The ConSpel version in that study used same baseline algorithms in combination with the *n*-grams-with-frequencies algorithm and Dejavu algorithm. However, the *n*-grams-with-frequencies in that study used a different method of integration – each size *n* of *n*-grams window was considered a separate ranker (similar to Stehouwer and van Zaanen, 2009). In present study we investigated a different method of integration – summing evidence from all overlapping *n*-grams and for all window sizes within the same ranker (as proposed by Bergsma *et al.*, 2009). This method of integration is clearly more effective (F1=84.32), more than 6% better than our previous result.

We have also considered using a measure of association instead of log counts. Using *n*-grams with positive normalized PMI provides results that are slightly better than using log counts. For the full set of 3,000 essays, the best results with PNPMI are achieved when using a window of up to four-grams – F1=84.77, and the best results with log counts are achieved when using a window of up to five-grams (the difference is not statistically significant). Both approaches show similar progression when longer *n*-grams are added – performance improves up to four-grams. When evaluation considers only the top correction candidate for each misspelling (for automatic correction), addition of five-grams is not helpful. However, when evaluation considers the top five correction candidates for a misspelling, addition of five-grams provides better results, but not statistically significant (as compared to using up to four-grams).

Overall, each additional order of *n*-grams provides a diminishing amount of improvement (see Figure 3), as if reaching a plateau. One possible explanation for this might be the coverage of *n*-grams in the database.[7,8] We have calculated how often the database returns numeric values (rather than "no data") for *n*-gram queries of different sizes. On average, the database returns valid data for 45% of bigram queries, 10% of trigram queries, 2% of four-gram queries and just 0.5% of five-gram queries. Note that the *n*-gram queries that are generated by ConSpel are often unnatural – many of them are "artificial *n*-grams" – a combination of a potential candidate correction with real context, so the large proportion of unfulfilled queries is quite expected. However, the pattern is suggestive – given the diminishing supply of evidence, the effectiveness of longer *n*-grams diminishes. It is quite surprising that with just 2% valid returns, the four-grams do provide a significant improvement for overall performance of *n*-gram-based algorithms. It is also quite surprising that with just 0.5% valid returns, the influence of five-grams is discernible (albeit not statistically significant).

Concerning the log-counts and PNPMI *n*-gram-based approaches, the similarity of their results (see Figure 3) may suggest that they are tapping exactly the same information. However, there is a surprising finding – when these two approaches are combined (see Table 14), the result provides additional improvement (1% over just PNPMI, 1.4% over just log counts), which is statistically significant. In addition, PNPMI works better than log-counts in the subset of high quality essays. Thus, it seems that PNPMI captures a slightly different aspect than frequency.

In this study we have also considered how the overall quality of an essay text influences contextual algorithms. The results are mixed. For the *n*-grams-with-log-counts algorithm, correction in lower quality essays (LQ) shows greater improvement (10.5%) than in higher quality (HQ) essays (8.7%), and the difference is significant ($p<.01$). For the *n*-grams-with-PNPMI, the amounts of improvement are closer: 10.67% for LQ and 9.79% for HQ, and the difference is significant ($p<.01$). Amounts of improvement are very close with the Dejavu algorithm (3.89% for LQ and 3.73 for HQ) and with Semantic Relatedness algorithm (3.92% for LQ and 3.59% for HQ). In both cases the differences between LQ and HQ are not significant. The Biasing algorithm shows a different pattern. When it works (in the subset of 750 essays), the improvement in the HQ set (7.16%) is greater than in the LQ set (4.47%), the difference is significant ($p<.01$). One possible explanation for this might be that HQ essays are more topically focused and possibly use more vocabulary from the prompt materials. This may be another opening for further research. Overall, *n*-grams-based-algorithms are the most effective, for both low and high quality essays and especially effective for lower quality essays. Semantic Relatedness and Dejavu are effective to a similar extent for both types of essays.

7. We thank an anonymous reviewer for pointing this out.
8. The database for current study was derived from Google Web1T. It contains 1.8 billion *n*-gram types of sizes 1-5. The derivation filtered out irrelevant data (e.g. Web and email addresses, non-English words, errors, etc.), as described in Flor (2013).

Interestingly, experiments have shown that Semantic Relatedness and Dejavu do not combine well together. Uncovering reasons for this may be subject for further research.

8. Conclusions

This article presented investigations of using four types of contexts for automatic correction of spelling errors in student essays, focusing on single-token non-word misspellings. We have described an implemented a state-of-the-art system, ConSpel, which allows a modular selection of algorithms for spelling correction. The task is framed as re-ranking of correction candidates. A baseline system includes algorithms that compute edit distance, phonetic similarity and word frequency as features for ranking candidates. Contextual algorithms can be engaged as additions to the baseline system, for improved, context-sensitive ranking of candidate spelling corrections. We have also presented experimental results of automatic error correction for a corpus of 3,000 student essays from high-stakes international English examinations and demonstrated that all four types of contexts are effective for improving the accuracy of error correction, as compared to a baseline that ranks correction candidates without context.

Three types of contexts in this investigation are internal to the text being corrected. The first type of context is the exact local context (word sequence) around the misspelling, which is captured via word *n*-grams. The experiments show that using *n*-gram frequencies (from a web-scale *n*-gram model) is effective for improving over the baseline results, and adding longer *n*-grams provides considerable improvement over using only short *n*-grams. We have also shown an alternative to using *n*-gram frequencies. Using a specific statistical association measure – positive normalized pointwise mutual information – over local *n*-grams, provides good improvement of correction accuracy relative to baseline, even slightly better than frequency. Moreover, combining the two methods provides additional improvement.

Another type of context is the (unordered) local lexical neighborhood of the misspelled token. We have presented a conjecture that the amount of semantic relatedness (or lexical cohesion) between each candidate and the lexical neighborhood of a misspelling can be useful for candidate ranking. This conjecture is similar to the idea of using lexical cohesion for correcting real-word misspellings (Budanitsky and Hirst, 2006). Although utilization of lexical cohesion for correction of non-word misspellings is a natural extension of that idea, to the best of our knowledge, we are the first to propose and implement it. In addition, rather than using a structured knowledge-based lexical resource, such as WordNet, we utilize a large scale distributional semantic model as wide-coverage resource for estimating semantic relatedness. The experiments show that using Semantic Relatedness (estimated with PNPMI association measure) provides about 3.8% improvement of

correction accuracy over the baseline. This result is a first empirical demonstration that Semantic Relatedness is applicable and effective for correction of non-word misspellings.

The third type of intra-textual context is word repetitions in text. A misspelled token in a text may be an instance of a word (type) that is used repeatedly in that text, and finding such possible repetitions (and inflectional variants) in the text can be useful for ranking correction candidates. Moreover, we have introduced a modified version of this approach, which can handle even cases when all (or most) of the occurrences of an intended word in a text are misspelled. The trick is that while ranking candidate corrections for one misspelled token, it can be useful to peek into the candidate lists of other misspelled tokens in the same text (global mutual optimization). The experiments show that using this "Dejavu" approach provides about 3.8% improvement of correction accuracy over the baseline results.

The fourth type of context is external to the text. If the topic of a text is known, or can be confidently estimated, this knowledge can be used to bias the error-correction towards the topic. Using a weak version of this conjecture, we biased error correction for essays, by using word lists from the prompts of the writing tasks. The experiments show that using this approach can provide about 5.3% improvement of correction accuracy over the baseline results, but only when the biasing context is rich enough.

All four types of contexts presented in this paper show significant improvement of automatic spelling correction as compared to baseline (non-contextual) algorithms. The baseline performance has F1=74.4%. Using *n*-grams-with-log-counts provides a 9.93% improvement over the baseline. Using *n*-grams-with-PNPMI provides 10.38% improvement over the baseline. Semantic Relatedness provides 3.8% improvement. Word-repetitions (Dejavu) algorithm also provides 3.8% improvement. Topical biasing (under suitable conditions) provides 5.3% improvement. Clearly, the *n*-grams-based approaches are the most effective and they provide the best improvement of performance.

We have also experimented with combining the various types of contexts. Overall, the best combination uses both *n*-gram-based algorithms and Semantic Relatedness and provides about 11.48% improvement over the baseline. With context-sensitive algorithms, the ConSpel system achieves 85.87% error correction accuracy in evaluation that uses the top-ranked candidate for each error. The system places the adequate correction among the five top-ranked candidates in 95% of the cases. This can be taken as an indication that there is a potential for additional improvements of automatic error-correction via re-ranking of correction candidates.

The main advantage of the described system is that while being effective it is also very generic – it uses general-purpose language models that are derived from huge masses of (mostly) correct English text. It works well without an error-model, and it can be easily tuned/biased to specific topics by providing rich word lists for those topics. Although the system uses Web-scale language models, its practical

deployment is not limited to high-end server platforms. Using the TrendStream database library (Flor, 2013), the ConSpel system runs locally on laptops and desktops.

The line of research presented in this article naturally extends to other types of misspellings. One area of research is correction of multi-token misspellings (Cucerzan and Brill, 2004). For some multi-token misspellings, such as *mor efun* (for "more fun"), each component can be corrected by itself. For many other multi-token misspellings (splits, such as *conten t*, for "content"), a system must consider the parts together, and that is not quite trivial when additional errors are present, e.g. *adittina lly*. Context-sensitive methods might prove to be quite useful in such task.

There is an obvious affinity of context sensitive methods for correction of both real-word and non-word misspellings. In this study, we have adapted the method proposed by Bergsma *et al.* (2009) for real-word misspellings, to non-word misspellings. For real-word spelling correction, Fossati and Di Eugenio (2007) have shown the usefulness of parts-of-speech contexts, and Xu *et al.* (2011) have shown the usefulness of dependency parsing. Similar approaches may be useful for correction of non-word errors. At the same time, we are working on applying the contextual methods presented in this paper toward detection and correction of real-word spelling errors.

Acknowledgements

Many thanks to Beata Beigman Klebanov, Yoko Futagi and Jana Sukkarieh, for valuable comments during preparation of the manuscript. This article has also benefited from the comments of anonymous reviewers.

9. References

Atkinson K., "GNU Aspell", 2011. Software available at http://aspell.net.

Baroni M., Lenci A., "Distributional Memory: A General Framework for Corpus-Based Semantics", *Computational Linguistics*, vol. 36, no. 4, p. 673-721.

Bentley J. L., Sedgewick R., "Fast Algorithms for Sorting and Searching Strings", *Proceedings of the 8th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA'97, 1997, New Orleans, LA, USA, p. 360-399.

Bergsma S., Lin D., Goebel R., "Web-Scale N-gram Models for Lexical Disambiguation", *Proceedings of International Joint Conference on Artificial Intelligence* (IJCAI-2009), 2009, p. 1507-1512.

Bestgen, Y., Granger, S., "Categorising Spelling Errors to Assess L2 Writing", *International Journal of Continued Engineering Education and Life-Long Learning*, vol. 21, no. 2/3, 2011, p. 235-252.

Bouma G., "Normalized (Pointwise) Mutual Information in Collocation Extraction", In: Chiarcos, Eckart de Castilho & Stede (eds), "Von der Form zur Bedeutung: Texte automatisch verarbeiten / From Form to Meaning: Processing Texts Automatically", *Proceedings of the Biennial GSCL Conference 2009*, Tübingen, Gunter Narr Verlag, p. 31-40.

Brants T., Franz A., "Web 1T 5-gram Version 1", LDC2006T13, 2006, Philadelphia, PA, USA: Linguistic Data Consortium.

Brill E., Moore R.C., "An Improved Error Model for Noisy Channel Spelling Correction", In *Proceedings of the 38th Annual Meeting of ACL*, 2000, p. 286-293.

Budanitsky A., Hirst G., "Evaluating WordNet-based Measures of Semantic Distance", *Computational Linguistics*, vol. 32, no. 1, 2006, p. 13-47.

Bullinaria J., Levy J, "Extracting Semantic Representations from Word Co-occurrence Statistics: A Computational Study", *Behavior Research Methods*, vol. 39, p. 510-526.

Carlson A., Fette I., "Memory-Based Context-Sensitive Spelling Correction at Web Scale", *Proceedings of the 6th International Conference on Machine Learning and Applications*, 2007, p. 166-171.

Chen Q., Li M., Zhou M., "Improving Query Spelling Correction Using Web Search Results", In *Proceedings of the 2007 Conference on Empirical Methods in Natural Language* (EMNLP-2007), p. 181-189.

Church K., Hanks P., "Word Association Norms, Mutual Information and Lexicography", *Computational Linguistics*, vol. 16, no. 1, 1990, p. 22-29.

Cook V., "L2 Users and English Spelling", *Journal of Multilingual and Multicultural Development*, vol. 18, no. 6, 1997, p. 474-488.

Crossley S.A., Salsbury T., McCarthy Ph., McNamara D.S., "Using Latent Semantic Analysis to Explore Second Language Lexical Development", In Wilson, D. and Chad Lane, H. (Eds.): *Proceedings of the 21st International Florida Artificial Intelligence Research Society Conference*, 2008, p. 136-141.

Cucerzan, S., Brill, E. "Spelling Correction as an Iterative Process that Exploits the Collective Knowledge of Web Users", *Proceedings Of Conference On Empirical Methods In Natural Language Processing* (EMNLP-2004), 2004, p. 293-300.

Damerau F., "A Technique for Computer Detection and Correction of Spelling Errors", *Communications of the ACM*, vol. 7, no. 3, 1964, p. 659-664.

De Felice R., Pulman S.G., "A Classifier-Based Approach to Preposition and Determiner Error Correction in L2 English", In *Proceedings of the 22nd International Conference on Computational Linguistics* (COLING 2008), 2008, p. 69-176.

Deorowicz S., Ciura M.G., "Correcting Spelling Errors by Modelling Their Causes", *International Journal of Applied Mathematics and Computer Science*, vol. 15, no. 2, 2005, p. 275-285.

Desmet Ch., Balthazor R., "Finding Patterns in Textual Corpora: Data Mining, Research, and Assessment in First-year Composition", Conference Proceedings, Computers and Writing 2005: *New Writing and Computer Technologies* (2005), p. 1-8.

Dikli S., "An Overview of Automated Scoring of Essays", *Journal of Technology, Learning, and Assessment*, vol. 5, no. 1, 2006, p. 4-35.

ETS. 2011a. GRE®: Introduction to the Analytical Writing Measure. Educational Testing Service. www.ets.org/gre/revised_general/prepare/analytical_writing (last accessed on September 14, 2012).

ETS. 2011b. TOEFL® iBT® Test Content. Educational Testing Service. www.ets.org/toefl/ibt/about/content (last accessed on September 14, 2012).

ETS, 2007, The Criterion® Teaching Guide: Using the Criterion Online Writing Evaluation Service for Differentiated Instruction in the College Classroom. Educational Testing Service, Princeton, NJ, USA.
http://www.ets.org/Media/Resources_For/Higher_Education/pdf/Criterion_Teacher_Guide_web_6487.pdf (last accessed on September 14, 2012).

Evert S., "Corpora and Collocations", In A. Lüdeling and M. Kytö (eds.), *Corpus Linguistics: An International Handbook*, 2008, article 58. Mouton de Gruyter: Berlin.

Flor M., "A Fast and Flexible Architecture for Very Large Word N-gram Datasets", *Natural Language Engineering*, vol. 19, no. 1, 2013, p. 61-93. DOI: http://dx.doi.org/10.1017/S1351324911000349

Flor M., Futagi Y., "Producing an Annotated Corpus with Automatic Spelling Correction." In S. Granger, G. Gilquin & F. Meunier (eds) *Twenty Years of Learner Corpus Research: Looking back, Moving ahead. Corpora and Language in Use* – Proceedings 1, 2013, Presses universitaires de Louvain: Louvain-la-Neuve, Belgium, p. 139-154.

Flor M., Futagi Y., "On Using Context for Automatic Correction of Non-word Misspellings in Student Essays", *The 7th Workshop on the Innovative Use of NLP for Building Educational Applications*, BEA-7 (at NAACL HLT 2012), Montreal, Canada, June 3-8, 2012, p. 105-115.

Fossati, D., Di Eugenio B., "A Mixed Trigrams Approach for Context Sensitive Spell Checking", *8th International Conference on Intelligent Text Processing and Computational Linguistics*, CICLing-2007, Mexico City, Mexico. February 2007.

Futagi Y., "The Effects of Learner Errors on the Development of a Collocation Detection Tool", In *Proceedings of the 4th Workshop on Analytics for Noisy Unstructured Text Data* (AND '10), 2010, p. 27-34.

Gao J., Li X., Micol D., Quirk Ch., Sun X., "A Large Scale Ranker-Based System for Search Query Spelling Correction", In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING 2010.

Golding A.R., Roth D., "A Winnow-based Approach to Context-sensitive Spelling Correction", *Machine Learning*, vol. 34, no. 1-3, 1999, p. 107-130.

Graff, D., and Cieri, C., "English Gigaword", 2003, Philadelphia, PA, USA: Linguistic Data Consortium. http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2003T05

Granger S., Wynne M., "Optimising Measures of Lexical Variation in EFL Learner Corpora", in Kirk, J. (Ed.): *Corpora Galore*, 1999, Rodopi, Amsterdam, p. 249-257.

Halliday M.A.K., Hasan R., "Cohesion in English", 1976, Longman: London.

Hoey M., “Patterns of Lexis in Text”, 1991, Oxford University Press.

Hoey M., “Lexical Priming: A New Theory of Words and Language”, 2005, Routledge.

Hovermale D.J., “An Analysis of the Spelling Errors of L2 English Learners”. Presented at CALICO 2010 Conference, Amherst, MA, USA, June 10-12, 2010. Available from http://www.ling.ohio-state.edu/~djh/presentations/djh_CALICO2010.pptx

Islam A., Inkpen D., “Real-word Spelling Correction Using Google Web1T N-gram with Backoff”, *Proceedings of the IEEE International Conference on Natural Language Processing and Knowledge Engineering* (IEEE NLP-KE’09), Dalian, China, September 2009, p. 1-8.

Kernighan M., Church K., Gale W., “A Spelling Correction Program Based on a Noisy Channel Model”, *Proceedings of the 13th Conference on Computational Linguistics* (COLING ’90), 1990, p. 205-210.

Kuhn R., De Mori R., “A Cache-based Natural Language Model for Speech Recognition”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 6, 1990, p. 570-583.

Kukich K., “Techniques for Automatically Correcting Words in Text”, *ACM Computing Surveys*, vol. 24, 1992, p. 377-439.

Landauer T.K., Laham D., Foltz P., “Automatic Essay Assessment”, *Assessment in Education*, vol. 10, no. 3, 2003, p. 295-308.

Leacock, C., Chodorow, M., Gamon M., Tetreault J., “Automated Grammatical Error Detection for Language Learners”, *Synthesis Lectures on Human Language Technologies*, no. 9, Morgan & Claypool, 2010, Princeton, USA.

Leacock C., Chodorow, M., “C-rater: Automated Scoring of Short-answer Questions”, *Computers and Humanities*, vol. 37, 2003, p. 389-405.

Levenshtein V., “Binary Codes Capable of Correcting Deletions, Insertions and Reversals”, *Soviet Physics Doklady*, 10, 1966, p. 707-710.

Lunsford A.A., Lunsford K.J., “Mistakes Are a Fact of Life: A National Comparative Study”, *College Composition and Communication*, vol. 59, no. 4, 2008, p. 781-806.

Manning, C., Schütze H., “Foundations of Statistical Natural Language Processing”, 1999, Cambridge, Massachusetts, USA: MIT Press.

Mays E., Damerau F., Mercer R., “Context Based Spelling Correction”, *Information Processing and Management*, vol. 23, no. 5, 1991, p. 517-522.

McCarthy D., Navigli R., “SemEval-2007 Task 10: English Lexical Substitution Task”, *Proceedings of Semeval-2007 Workshop* (SEMEVAL) in the 45th Annual Meeting of the Association for Computational Linguistics (ACL 2007), June 23-24, 2007, Prague, Czech Republic, p. 48-53.

Mitton R., “Ordering the Suggestions of a Spellchecker Without Using Context”, *Natural Language Engineering*, vol. 15, no. 2, 2008, p. 173-192.

Mitton R., “English Spelling and the Computer”, 1996, Harlow, Essex: Longman Group. Available electronically from http://eprints.bbk.ac.uk/469

Mitton R., Okada T., "The Adaptation of an English Spellchecker for Japanese Writers", presented at the Symposium on Second Language Writing, September 15-17, 2007, Nagoya, Japan. Available electronically from http://eprints.bbk.ac.uk/592

Mohammad S., Hirst G. "Distributional Measures of Concept-Distance: A Task-oriented Evaluation", In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing* (EMNLP 2006), p. 35-43.

Morris J, Hirst G., "Lexical Cohesion Computed by Thesaural Relations as an Indicator of the Structure of Text", *Computational Linguistics*, vol. 17, no. 1, 1991, p. 21-48.

Nagata, R., Whittaker E., Sheinman V., "Creating a Manually Error-tagged and Shallow-parsed Learner Corpus", In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics* (EMNLP-2011), p. 1210-1219, Portland, Oregon, USA.

Okada T., "Spelling Errors Made by Japanese EFL Writers: With Reference to Errors Occurring at the Word-initial and the Word-final Position", In V. Cook and B. Bassetti (Ed.), *Second Language Writing Systems*, 2005, Clevedon: Multilingual Matters, p. 164-183.

Pecina P., "Lexical association measures and collocation extraction", *Language Resources & Evaluation*, vol. 44, 2010, p.137-158.

Pérez D., Alfonseca E., Rodríguez P., "Application of the Bleu Method for Evaluating Free-text Answers in an E-learning Environment", In *Proceedings of the Language Resources and Evaluation Conference* (LREC-2004), 2004, p. 1351-1354.

Philips L., "The Double-metaphone Search Algorithm", *C/C++ User's Journal*, June, 2000.

Quinlan T., Higgins D., Wolff S. "Evaluating the Construct-Coverage of the e-rater® Scoring Engine", Research Report RR-09-01, 2009, Educational Testing Service, Princeton, NJ, USA. Available online: http://www.ets.org/Media/Research/pdf/RR-09-01.pdf

Ramineni C., Trapani C.S., Williamson D.M., Davey T., Bridgeman B., "Evaluation of the e-rater® Scoring Engine for the GRE® Issue and Argument Prompts", Research Report RR-12-02, 2012, Educational Testing Service, Princeton, NJ, USA. Available online: http://www.ets.org/research/policy_research_reports/rr-12-02

Ramineni C., Trapani C.S., Williamson D.M., Davey T., Bridgeman B., "Evaluation of the e-rater® Scoring Engine for the TOEFL® Independent and Integrated Prompts", Research Report RR-12-06, 2012, Educational Testing Service, Princeton, NJ, USA. Available online: http://www.ets.org/research/policy_research_reports/rr-12-06

Ristad E.S., Yianilos, P.N., "Learning String Edit Distance", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 5, 1998, p. 522-532.

Rozovskaya A., Sammons M., Roth D., "The UI System in the HOO 2012 Shared Task on Error Correction", *The 7th Workshop on the Innovative Use of NLP for Building Educational Applications*, BEA-7, Montreal, Canada, June 3-8, 2012, p. 272-280.

Stehouwer H, van Zaanen M., "Language Models for Contextual Error Detection and Correction", *Proceedings of the EACL-2009 Workshop on Computational Linguistic Aspects of Grammatical Inference*, Athens, Greece, March 30, 2009, p. 41-48.

Strohmaier C., Ringlstetter C., Schulz K., Mihov S., "Lexical Postcorrection of OCR-results: The Web as a Dynamic Secondary Dictionary?", In *Proceedings of the 7th International Conference on Document Analysis and Recognition*, ICDAR '03, 2003, p. 1133.

Sukkarieh J.Z., Blackmore J., "C-rater: Automatic Content Scoring for Short Constructed Responses", In *Proceedings of the 22nd International Florida Artificial Intelligence Research Society Conference*, 2009, p. 290-295.

Tong X., Evans D.A., "A Statistical Approach to Automatic OCR Error Correction in Context", Proceedings of the Fourth Workshop on Very Large Corpora, August 4, 1996, Copenhagen, Denmark, p. 88-100.

Turney P.D., Pantel P., "From Frequency to Meaning: Vector Space Models of Semantics", *Journal of Articial Intelligence Research*, 37, 2010, p. 141-188.

Warschauer M., Ware P., "Automated Writing Evaluation: Defining the Classroom Research Agenda", *Language Teaching Research*, vol. 10, no. 2, 2006, p.157-180.

Whitelaw C., Hutchinson B., Chung G.Y., Ellis G., "Using the Web for Language Independent Spellchecking and Autocorrection", *Proceedings Of Conference On Empirical Methods In Natural Language Processing* (EMNLP-2009), Singapore, 2009, p. 890-899.

Wick M., Ross M., Learned-Miller E., "Context-Sensitive Error Correction: Using Topic Models to Improve OCR", *Proceedings of the 9th International Conference on Document Extraction and Analysis*, ICDAR '07, 2007, p. 1168-1172.

Wilcox-O'Hearn A., Hirst G., Budanitsky A., "Real-word Spelling Correction with Trigrams: A Reconsideration of the Mays, Damerau, and Mercer Model", In *Proceedings of CICLing-2008*, 2008, p. 605-616.

Xu W., Tetreault J., Chodorow M., Grishman R., Zhao L., "Exploiting Syntactic and Distributional Information for Spelling Correction with Web-Scale N-gram Models", *Proceedings Of Conference On Empirical Methods In Natural Language Processing*, EMNLP-2011, p. 1291-1300.

Yu G., "Lexical Diversity in Writing and Speaking Task Performances", *Applied Linguistics*, vol. 31, no. 2, p. 236-259.

Zhang Y., He P., Xiang W., Li M., "Discriminative Reranking for Spelling Correction", *Proceedings of the 20th Pacific Asia Conference on Language, Information and Computation*, 2006, p. 64-71.

Zhang Z., Gentile A.L., Ciravegna F., "Recent Advances in Methods of Lexical Semantic Relatedness – a Survey", *Natural Language Engineering*, 2012, available online, DOI: http://dx.doi.org/10.1017/S1351324912000125