Exploiting Syntactic and Distributional Information for Spelling Correction with Web-Scale N-gram Models

Wei Xu[c,*] **Joel Tetreault**[a] **Martin Chodorow**[b] **Ralph Grishman**[c] **Le Zhao**[d]
[a]Educational Testing Service, Princeton, NJ, USA
jtetreault@ets.org
[b]Hunter College of CUNY, New York, NY, USA
martin.chodorow@hunter.cuny.edu
[c]New York University, NY, USA
{xuwei,grishman}@cs.nyu.edu
[d]Carnegie Mellon University, Pittsburgh, PA, USA
lezhao@cs.cmu.edu

Abstract

We propose a novel way of incorporating dependency parse and word co-occurrence information into a state-of-the-art web-scale n-gram model for spelling correction. The syntactic and distributional information provides extra evidence in addition to that provided by a web-scale n-gram corpus and especially helps with data sparsity problems. Experimental results show that introducing syntactic features into n-gram based models significantly reduces errors by up to 12.4% over the current state-of-the-art. The word co-occurrence information shows potential but only improves overall accuracy slightly.

1 Introduction

The function of context-sensitive text correction is to identify word-choice errors in text (Bergsma et al., 2009). It can be viewed as a lexical disambiguation task (Lapata and Keller, 2005), where a system selects from a predefined confusion word set, such as {affect, effect} or {complement, compliment}, and provides the most appropriate word choice given the context. Typically, one determines if a word has been used correctly based on lexical, syntactic and semantic information from the context of the word. One of the top performing models of spelling correction (Bergsma et al., 2010) is based on web-scale n-gram counts, which reflect both syntax and meaning. However, even with a large-scale n-gram corpus, data sparsity can hurt performance in two ways. First, n-gram based methods require exact word and order matches. If there is a low frequency word in the context, such as a person's name, there will be little, if any, evidence in the n-gram data to support the usage. Second, if the target confusable word is rare, there will not be enough n-gram support or training data to render a confident decision. Because of the data sparsity problem, language modeling is not always sufficient to capture the meaning of the sentence and the correct usage of the word.

Take a sentence from The New York Times (NYT) for example: "'This fellow's won a war,' the dean of the capital's press corps, David Broder, announced on 'Meet the Press' after **complimenting** the president on the 'great sense of authority and command' he exhibited in a flight suit." Unfortunately, neither the phrase "complementing the president" nor "complimenting the president" exists in the web-scale Google N-gram corpus (Brants and Franz, 2006). The n-gram models decide solely based on the frequency of the bi-grams "after comple(i)menting" and "comple(i)menting the", which are common usages for both words. The real question is whether we are more likely to "compliment" or "complement" a person, the "president". Several clues could help us answer that question. A dependency parser can identify the word "president" as the subject of "compliment" or "complement" which also may be the case in some of the training data. Lexical co-occurrence (Edmonds, 1997) and semantic word relatedness measurements, such as Random Indexing (Sahlgren, 2006), could provide evidence that "compliment" is more likely to co-occur with "president" than "complement". Fur-

[*] This work was done when the first author was an intern for Educational Testing Service.

Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pages 1291–1300,
Edinburgh, Scotland, UK, July 27–31, 2011. ©2011 Association for Computational Linguistics

thermore, some important clues can be quite distant from the target word, e.g. outside the 9-word context window Bergsma et al. (2010) and Carlson (2007) used. Consider another sentence in the NYT corpus, “GM says the addition of OnStar, which includes a system that automatically notifies an OnStar operator if the vehicle is involved in a collision, **complements** the Vue’s top five-star safety rating for the driver and front passenger in both front- and side-impact crash tests.” The dependency parser finds the object of “complement” is “rating”, which is outside the 9-word window.

We propose enhancing state-of-the-art web-scale n-gram models for spelling correction with syntactic structures and distributional information. For our work, we build on a baseline system that combines n-gram and lexical features (Bergsma et al., 2010). Specifically, this paper makes the following contributions:

1. We show that the baseline system can be improved by augmenting it with dependency parse features.
2. We show that the impact of parse features can be further augmented when combined with distributional information, specifically word co-occurrence information.

In the following section, we describe related work and how our approach differs from these approaches. In Sections 3 and 4, we discuss our methods for using parse features and word co-occurrence information. In Section 5, we present experimental results and analysis.

2 Related Work

A variety of approaches have been proposed for context-sensitive spelling correction ranging from semantic methods to machine learning classifiers to large-scale n-gram models.

Some semantics-based systems have been developed based on an intuitive assumption that the intended word is more likely to be semantically coherent with the context than is a spelling error. Jones and Martin (1997) made use of the semantic similarity produced by Latent Semantic Analysis. Budanitsky and Hirst (2001) investigated the effectiveness of predicting words based on different semantic similarity/distance measures in WordNet. Both systems report performance that is lower than systems developed more recently.

A variety of machine-learning methods have been proposed in spelling correction and preposition and article error correction fields, such as Bayesian classifiers (Golding, 1995; Golding and Roth, 1996), Winnow-based learning (Golding and Roth, 1999), decision lists (Golding, 1995), transformation-based learning (Mangu and Brill, 1997), augmented mixture models (Cucerzan and Yarowsky, 2002) and maximum entropy classifiers (Izumi et al., 2003; Han et al., 2006; Chodorow et al., 2007; Tetreault and Chodorow, 2008; Felice and Pulman, 2008). Despite their differences, these approaches mainly use contextual features to capture the lexical, semantic and/or syntactic environment of the target word.

The use of distributional similarity measures for spelling correction has been previously explored in (Mohammad and Hist, 2006). In our work, distributional similarity is not the primary contribution but we show the impact it can have when used in conjunction with a large scale n-gram model and with parse features, which allows the system to select words outside the local window for distributional similarity. In the prior work, the words for distributional similarity are constrained to the local window, and positional information of the words is not encoded.

Recent work (Carlson and Fette, 2007; Gamon et al., 2008; Bergsma et al., 2009) has demonstrated that large-scale language modeling is extremely helpful for contextual spelling correction and other lexical disambiguation tasks. These systems make the word choice depending on how frequently each candidate word has been seen in the given context in web-scale data. As n-gram data has become more readily available, such as the Google N-gram Corpus, the likelihood of a word being used in a certain context can be better estimated.

Bergsma et al. (2009; 2010) presented a series of simple but powerful models which relied heavily on web-scale n-gram counts. From the Google Web N-gram Corpus, they retrieve counts of n-grams of different sizes (2-5) and positions that span the target word w0 within a window of 9 words. For example, for the following sentence: “The system tried to decide {among, between} the two confusable words.”, the method would extract the five 5-gram patterns, shown below in Figure 2, where $w0$ can be either word in the confusion set {among, between} in this particular example. Similarly, there are four 4-grams, three 3-grams, and two 2-grams, in total, 14 n-grams for each of the words in the confusion set.

system tried to decide $w0$

tried to decide $w0$ the

to decide $w0$ the two

decide $w0$ the two confusable

$w0$ the two confusable words

We briefly describe three of Bergsma et al.’s (2009; 2010) best systems below, which are reported to achieve state-of-the-art accuracy (NG = n-gram; LEX = lexical).

1. **sumLM**: For each candidate word, (Bergsma et al., 2009) sum the log-counts of all 14 patterns filled with the candidate, and choose the candidate with the highest total.

2. **NG**: Bergsma et al. (2009) exploit each candidate’s 14 log-counts of n-gram patterns as features in a Support Vector Machine (SVM) model.

3. **NG+LEX**: Bergsma et al. (2010) augment the NG model with lexical features (described in detail in Section 3.1).

Bergsma et al. (2009; 2010) restricted their experiments to only five confusion sets where the reported performance in (Golding and Roth, 1999) was below 90%: {among, between}, {amount, number}, {cite, sight, site}, {peace, piece} and {raise, rise}. They reported that the SVM model with NG features outperformed its unsupervised version, sumLM. However, the limited confusion word sets they evaluated may not comprehensively represent the word usage errors that writers typically make. In this paper, we test nine additional commonly confused word pairs to expand the scope of the evaluation. These words were selected based on their lower frequencies compared to the five pairs in the above work (as shown later in Table 2).

3 Enhanced N-gram Models with Parse Features

To our knowledge, only (Elmi and Evans, 1998) have used parsing for spell correction. They focus on using a parser as a filter to discriminate between possible real-world corrections where the part-of-speech differs. In our work, we show that parse features are effective when used directly in the classification mode (as opposed to as a final filter) to select the best correction regardless of whether or not the part-of-speech of the choices differ.

Statistical parsers have also seen limited use in the sister tasks of preposition and article error detection (Hermet et al., 2008; Lee and Knutsson, 2008; Felice and Pulman, 2009; Tetreault et al., 2010) and verb sense disambiguation (Dligach and Palmer, 2008). In those instances where parsers have been used, they have mainly provided shallow analyses or relations involving specific target words, such as a preposition or verb. Unlike preposition errors, spelling errors can occur in any word.

In this paper, we propose a novel way to incorporate the parse into spelling correction, applying the parser to sentences filled by each candidate word equivalently and extracting salient features. This overcomes two problem in the existing methods: 1) the parse trees of the same sentence filled by different confusion words can be different. However, in the test phase, we do not know which word should be put in the sentences to create parse features for test examples. Previous studies (Tetreault et al., 2010) failed to discuss this issue. 2) Some existing work (Whitelaw et al., 2009; Rozovskaya and Roth, 2010) in the text correction field introduced artificial errors into training data to adapt the system to better handle ill-formed text. But this method will encounter serious data sparsity problems when facing rare words.

3.1 Baseline System

We chose one of the leading spelling correction systems, (Bergsma et al., 2010), as our primary baseline. As noted earlier, it is an SVM-based system combining web-scale n-gram counts (NG) and contextual words (LEX) as features. To simplify the explanation, throughout the paper, we will only consider the situation with two confusion words. The problem with more than two words in pre-defined confusion sets can be solved similarly by using a one-vs.-all strategy. As we mentioned in Section 2, NG features include log-counts of 3-to-5-gram patterns for each candidate word with the given context. LEX features can be broken down into three subcategories: 1) bag-of-words (words at all positions in a 9-word window around the target word), 2) indicators for the words preceding or following the target word, and 3) indicators for all n-grams and their positions. For the sentence "The system tried to decide {among, between} the two confusable words.", examples of bag-of-word features would be "tried", "two", etc., the two positional bigrams would be "decide" and "the", and examples of the n-gram features would be right-trigram = "among the two" and left-4-gram = "tried to decide between".

3.2 Parse Features

The benefit of introducing dependency parse features is that 1) parse features capture contextual information in a larger context window; 2) parse features specify which words in the context are salient to the usage of the target word while purely lexically based approaches treat all words in the context equally. We use the Stanford dependency parser (de Marneffe et al., 2006) to extract six relevant feature classes.

Parse Features (PAR):

1. relation names (target word as head)
2. complement of the target word
3. combination of 1 and 2
4. relation names (target word as complement)
5. head of the target word
6. combination of 4 and 5

Each of these six classes of PAR features can contain zero to many values, since the target word can be involved in none to multiple grammatical relations and features of different filler words are merged together. The PAR features, like the LEX features, are binary. In Table 1, we present the parse features for an example sentence. The parse features here are listed as string values, but are later converted into binary numbers in the vectors for the SVM model.

4 Distributional Word Co-occurrence

Though lexical and parse features are complementary to n-gram models, they are learned from a normal training corpus and may not have enough coverage due to data sparsity. Take a sentence from the NYT for example: "An economist, he began his career as a professor – he is still called 'the professor,' by friends as a compliment and by foes as an insult – and taught at Harvard and Stanford ." If the most indicative word "friends" does not appear or does not appear enough times in the local context or dependencies with "compliment" as compared to "complement" in the training corpus, then the classifier may be unable to make the correct selection.

It is impractical and computationally costly to enlarge the training corpus without limit to include all possible language phenomena. A good compromise is to use word co-occurrence information from web-scale data. The other option is to make use of high-order word co-occurrence, which is included in many semantic word relatedness measures, such as Latent Semantic Analysis (LSA) (Landauer et al., 1998; Deerwester et al., 1990) or Random Indexing, both of which can be estimated from a moderate-size corpus.

Our intuition is to choose the confusion word which is most relevant to a given context. We define the salient words in context as a set M=m1, m2, m3, ..., and the relevance between two words as a function Relevance(w1, w2), which can either be calculated from word co-occurrence or Random Indexing. The score of each candidate word c in the confusion set given a context with meaningful words M is calculated by the following formula:

$$Score(c) = \sum_{m \in M} Relevance(c, m)$$

In this paper, we experiment with first-order word co-occurrence and Random Indexing as relevance measures. And we define salient contextual words as heads or complements in the dependency relations with the target word. In this way, we use the parse information to constrain the two distribution models. Thus the word co-occurrence information

Feature Name	PAR Features (compliment)	PAR Features (complement)
1. Head Relation Name	ccomp	appos
2. Head of Relation	says	collisions
3. Head Combination	ccomp_says	appos_collisions
4. Comp Relation Name	nsubj	dep
5. Comp of Relation	addition	rating
6. Comp Combination	nsub_addition	dep_rating

Table 1: Parse Feature Example for the sentence: "GM says the addition of OnStar, which includes a system that automatically notifies an OnStar operator if the vehicle is involved in a collision, **complements** the Vue's top five-star safety rating for the driver and front passenger in both front- and side-impact crash tests."

considerably overlaps with some values of the PAR features, but provides extra evidence from web-scale data rather than a limited amount of training data.

4.1 First-order Word Co-occurrence

The relevance based on first-order word co-occurrence is calculated from the Google Web 5-gram Corpus in a fashion similar to how we dealt with n-gram counts in the previous section. Given two words, w1 and w2, we consider all 8 possible patterns that appear in a local context (5-word window), where we use wildcard (*) to indicate any token:

```
w1 w2
w1  *  w2
w1  *   *  w2
w1  *   *   *  w2
w2 w1
w2  *  w1
w2  *   *  w1
w2  *   *   *  w1
```

The relevance is then calculated by summing the logarithm of each of the 8 different counts. Finally, we compare the score of each candidate word and output the one with higher score.

4.2 Random Indexing

The relevance scores based on Random Indexing are provided by a tool FRanI (Higgins, 2004) and a model trained on the Touchstone Applied Science Associates (TASA) corpus which contains 750k sentences and covers diverse topics (from a diversity of textbooks up to the college level). Take the sentence at the beginning of this section for example, where only the words "a" and "friends" are related to the target word (either "complement" or "compliment") by either relevance measure. The relevance based on Random Indexing for (complement, friends) is 0.08, (compliment, friends) is 0.19 and both (compliment, a) and (complement, a) are 0 because "a" is in the stop word list. Meanwhile, the relevance based on first order word co-occurrence for (compliment, friends) is 7.39, (complement, friends) is 5.38, (compliment, a) is 13.25, and (complement, a) is 13.42. The system with either kind of relevance outputs "compliment".

4.3 System Combination

Since the numeric measurement of word co-occurrence is not as specific as the PAR features and less trustworthy, adding word co-occurrence information as features into the classifier along with n-gram counts, lexical and parse features will hurt the overall performance. It is more practical to combine the two approaches in the following fashion:

1. When the SVM classifier (using NG, LEX and PAR features) has high confidence (over a certain threshold) in the output label, output that label;

2. Otherwise, output the results of the word relatedness/co-occurrence-based system.

5 Evaluation

We evaluate the effectiveness of syntactic and distributional information on spelling correction. The performance of the system is measured by accuracy: the percentage of sentences in the test data for which the system chooses the correct word. We compare our results against two baselines: 1) MAJOR chooses the most frequent candidate from the confusion set in the training corpus, and 2) Bergsma et al.'s (2010) best systems, NG+LEX. We include inflectional variants ("-ing", "-ed", "-s", "-ly") of confusion words in the evaluation, such as complementing, complimenting in addition to complement, compliment, because this better corresponds to the range of errors that may be encountered in actual use and thus increases the scope of the system as a real world application. Also following Bergsma et al. (2010), we use a linear SVM, more exactly, the L2-regularized L2-loss dual SVM in LIBLINEAR (Fan et al., 2008). Unlike Bergsma et al., who used development data to optimize parameters, we always use default parameters, since training data is limited for many of the words we are dealing with.

5.1 Data

Following Bergsma et al. (2009; 2010), the test examples are extracted from The New York Times (NYT) portion of Gigaword[1], but constrained to a 9-month publication time frame from October 2005 to July 2006. Unlike Bergsma et al. who use the same source as training data for the lexical features, our training data (for both lexical and parse features) comes from larger and more diverse news sources. We use the very large database from Sekine's n-gram search engine (Sekine, 2008) as training data, which consists of 1.9B words of newspaper text spanning 89 years from NYT, BBC, WSJ, Xinhua, etc.

We evaluate our systems on 5 confusion sets from Bergsma et al. (2009; 2010) and 9 commonly confused word pairs with moderate frequency in daily usage (randomly selected from those listed in English educational resources[2]). Shown in Table 2, these 9 sets of words appear much less frequently than the words selected by Bergsma et al., even given the fact that we are using a considerably large training corpus.

For each confusable word pair, sentences that contain either of the words are extracted to form training and test data. The word that appears in the original sentences of the news article is treated as the gold standard. For frequently occurring confusion word sets used by Bergsma et al., we extract up to 10k examples for testing, and up to 100k examples for training. For the 9 less frequent confusion word sets, we extract all the unique examples for training and testing from the above sources. The spelling correction system is evaluated by measuring its accuracy in comparison to the gold standard in test data. The error rate is the complement of accuracy.

Word Confusion Set	# in Training Corpus
adverse / averse	13.5k / 1.8k
advice / advise	62.k / 12.9k
allusion / illusion	1.0k / 5.4k
complement / compliment	6.8k / 3.1k
confidant / confident	2.4k / 63.6k
desert / dessert	24.7k / 3.7k
discreet / discrete	0.7k / 2.4k
elicit / illicit	1.9k / 10.0k
stationary / stationery	2.5k/2.3k
wander / wonder	3.3k / 39.5k

Table 2: Training Data Sizes for Common ESL Confused Words

Following Carlson et al. (2007) and Bergsma et al. (2009; 2010), we obtain the n-gram counts from the Google Web 1T 5-gram Corpus (Brants and Franz, 2006).

5.2 Experimental Results

We present the results for each set separately because each set may behave very differently, depending upon its frequency, part-of-speech, number of senses and other differences between the words in each confusion set. The overall accuracy across confusion sets is also presented to show the effectiveness of different approaches. The results are tested for statistical significance using McNemar's test of correlated proportions. The performance differences are marked as significant when $p < 0.05$.

5.2.1 Effectiveness of Parse Features

We exploit the n-gram counts (NG), lexical features (LEX) of Bergsma et al. (2010) and our own parse features (PAR) in linear SVM models.

The first comparison is between the supervised learning systems with LEX and LEX+PAR. As shown in Table 3, by exploiting our unique parse features, for the total 14 confusion sets, the accuracy increases on 12 sets and decreases on 2 sets. Overall, the spelling correction accuracy improves an ab-

[1] Available from the LDC as LDC2003T05

[2] Such as an English learning blog post at http://elisaenglish.pixnet.net/blog/post/1335194

solute 1.35% for our 9 confusion sets and 0.60% for Bergsma et al.'s 5 confusion sets.

The second comparison is to see how parse features interact with n-gram count features in a supervised classifier. The best system from (Bergsma et al., 2010) is listed in the table as "NG+LEX". As shown in Table 3, the parse features proved to be beneficial when augmenting this baseline, except for the decrease in accuracy on adverse, averse by only 2 cases out of 368, and among, between by 2 cases out of 10227. For all other confusion sets, parse features decrease the error rate by as much as 2.74% (absolute) and as much as 38.5% (relative). Improvements are statistically significant on all confusion sets together, although for each separate set, improvements are significant on only 5 sets, in part due to an insufficient number of test cases.

The reason that parse features are occasionally not helpful is because they sometimes include an uncommon word in dependencies, which happens to appear once with the wrong word but not with the correct word in the training data; or they sometimes include too common words, which bias the classifier in favor of the more frequent word in the confusion set. We also noticed that lexical features are not always helpful when added to n-gram count features, even for in-domain applications (i.e., with training data and test data coming from the same domain or corpus), as marked by underlines. However, lexical and parse features together show more significant and constant improvement over n-gram count-based models, as marked by α.

Of the six systems, every system that uses parse features gets the example correct in Section 1, "complementing the president"; LEX by itself also gets the example correct, but NG and NG+LEX fail.

In summary, our system NG+LEX+PAR outperforms the state-of-the-art system NG+LEX. It reduces the error rate by 12.4% across our 9 confusion sets and by 8.4% across Bergsma et al.'s 5 confusion sets. Both improvements are significant ($p < 0.05$) by the McNemar test. In addition, while NG+LEX is not always better than NG, NG+LEX+PAR is consistently better than NG.

5.2.2 Impact of Word Co-occurrence

The LIBLINEAR tool does not provide probability estimates for SVM models but Logistic Regression can. In this set of experiments, we train a Logistic Regression model with NG+LEX+PAR features and empirically set the confidence threshold at 0.6, as described in Section 4, based on the performance on two word pairs. In the combined system, when the Logistic Regression model estimates a probability higher than the threshold we output its results, otherwise we output the result of the system based on word co-occurrence.

Surprisingly, although Random Indexing takes into account more information than first-order word co-occurrence, it lowered overall performance substantially. Thus in Table 4, we only present results of using first-order word co-occurrence rather than Random Indexing. For all 12 confusion sets, distributional word co-occurrence information improves 9 sets and hurts 5 sets. Overall, it reduces the error rate slightly by 0.2% for our 9 sets and 1.5% for Bergsma et al.'s sets.

We believe there are two reasons why Random Indexing fared worse than first-order word co-occurrence: 1) Random Indexing considers co-occurrence on a document level, while our first-order word co-occurrence is limited to a 5-word window context. The latter is more suitable to context-sensitive spelling correction. 2) The model for Random Indexing is trained on a relatively small size corpus compared to the web-scale data we used to get n-gram count features for the classifier and thus is not able to introduce much new evidence besides the information carried by NG+LEX+PAR features.

Reason 2) also suggests why first-order co-occurrence helps on some occasions while not on other occasions. Its impact is limited because the word co-occurrence information overlaps with some of the PAR feature values as mentioned earlier. It improves some cases because it provides some new evidence from web-scale data to the system based on NG+LEX+PAR features. It introduces new errors because it simply favors the word that co-occurred more often regardless of other factors. Its impact is also limited because it is only considered when classifiers with NG+LEX+PAR features are not confident.

CONFUSION SET	# TEST	MAJOR	LEX	LEX+PAR	NG	NG+LEX	NG+LEX+PAR (&)
9 commonly cited ESL confusion pairs							
adverse / averse	368	85.87	**97.01**	96.74	91.03	**97.55**	97.01 (+22.2%) α
allusion / illusion	535	76.64	91.22	**91.40**	91.40	92.52	**93.08 (-7.5%)** α
complement / compliment	860	51.51	83.84	**85.12**	88.49	88.37	**89.53 (-10.0%)**
confidant / confident	2416	94.41	97.97	**98.30**	98.51	99.05	**99.09 (-4.3%)** α
desert / dessert	2357	70.81	90.71	**91.56**	87.31	93.68	**94.57 (-14.1%)** α*
discreet / discrete	219	79.45	84.48	**85.84**	85.84	90.41	**91.32 (-9.5%)** α
elicit / illicit	563	53.46	82.77	**95.56**	97.51	97.51	**98.22 (-28.6%)**
stationary / stationery	182	62.64	87.36	**92.31***	93.96	92.86	**95.60 (-38.5%)**
wander / wonder	6506	86.37	96.42	**97.42***	97.56	98.23	**98.48 (-13.9%)** α*
Total	13972	81.08	93.94	**95.29***	94.82	96.56	**96.99 (-12.4%)** α*
5 Original Bergsma pairs							
# among / between	10227	57.46	**91.89**	91.86	88.34	**93.60**	93.58 (+3.1%) α
# amount / number	7398	76.44	92.34	**93.16***	93.03	93.42	**94.08 (-10.1%)** α*
# cite / site	10185	95.71	99.42	**99.53**	99.16	99.52	**99.63 (-22.4%)**α
# peace / piece	7330	56.81	95.01	**97.01***	95.55	96.74	**97.46 (-22.2%)**α *
# raise / rise	9464	55.98	96.12	**96.64***	94.45	96.68	**97.05 (-11.5%)** α
Total	44604	68.92	95.09	**95.69***	94.07	96.09	96.42 (-8.4%) α

Table 3: Spelling correction precision (%), impact of adding parse features
SVM trained on 1G words of news text, tested on 9-months of NYT data.
*: Improvement of (NG+)LEX+PAR vs. (NG+)LEX is statistically significant.
α: Improvement of NG+LEX+PAR vs. NG is statistically significant.
&: Relative increase or decrease of error rate compared to ”NG+LEX”
#: As in Bergsma et al. (2009; 2010) no morphological variants of the words are used in evaluation

CONFUSION SET	# TEST	MAJOR	CLASSIFIER	COMBINED SYSTEM (&)
9 commonly cited ESL confusion pairs				
adverse / averse	368	85.87	**97.55**	96.74 (+33.3%)
allusion / illusion	535	76.64	**92.34**	**92.34 (- 0.0%)**
complement / compliment	860	51.51	89.88	**90.81 (-9.2%)**
confidant / confident	2416	94.41	**99.13**	99.05 (+9.5%)
desert / dessert	2357	70.81	93.98	**94.23 (-3.7%)**
discreet / discrete	219	79.45	90.41	**91.78 (-14.3%)**
elicit / illicit	563	53.46	98.40	**98.76 (-22.2%)**
stationary / stationery	182	62.64	93.41	**93.96 (-9.1%)**
wander / wonder	6506	86.37	**98.49**	98.36 (+9.2%)
5 Original Bergsma pairs				
# among / between	10227	57.46	92.73	**92.73 (-0.1%)**
# amount / number	7398	76.44	93.44	**93.76 (-4.74%)**
# cite / site	10185	95.71	**99.49**	99.47 (+3.8%)
# peace / piece	7330	56.81	96.19	**96.38 (-5.0%)**
# raise / rise	9464	55.98	**96.66**	96.59 (+2.2%)

Table 4: Spelling correction accuracy (%), impact of combining word co-occurrence
CLASSIFIER: Logistic Regression trained on 1G words of news text, tested on 9-months NYT data.
COMBINED SYSTEM: CLASSIFER plus system based on first-order word co-occurrence.
&: Relative increase or decrease in error rate compared to CLASSIFIER
#: As in Bergsma et al. (2009; 2010), no morphological variants of the words are used in evaluation

6 Conclusions

We propose a novel approach that uses parse features and lexical features together to improve the performance of web-scale n-gram models for spelling correction. This method is especially adaptive when less training data are available, which is the case for confusable words that are not very frequently used. We also investigate the effectiveness of incorporating web-scale word co-occurrence and corpus-based semantic word relatedness (Random Indexing).

For future work, we will investigate using semantic information (e.g. WordNet) to extend n-gram models. It will be interesting to see if the usage of the word "compliment" in "complimenting the president" can be estimated by considering similar usages in the corpus, such as "complimenting the student" or by creating an n-gram database of synset patterns. We will investigate extending, to other applications, this general methodology combining distributional, semantic and syntactic information with language models.

Acknowledgments

We wish to thank Michael Flor of Educational Testing Service for his TrendStream tool, which provides fast access and easy manipulation of the Google N-gram Corpus. We also thank Derrick Higgins of Educational Testing Service for his Random Indexing support. We also thank Satoshi Sekine of New York University, Matthew Snover of City University of New York, and Jing Jiang of Singapore Management University for their advice.

References

Shane Bergsma, Dekang Lin, and Randy Goebel. 2009. Web-scale n-gram models for lexical disambiguation. In *IJCAI*.

Shane Bergsma, Emily Pitler, and Dekang Lin. 2010. Creating robust supervised classifiers via web-scale n-gram data. In *ACL*.

Thorsten Brants and Alex Franz. 2006. Web 1T 5-gram version 1. Available at http://www.ldc.upenn.edu.

Alexander Budanitsky and Graeme Hirst. 2001. Semantic distance in wordnet: An experimental, application-oriented evaluation of five measures. In *ACL Workshop on WordNet and Other Lexical Resources*.

Andrew Carlson and Ian Fette. 2007. Memory-based context sensitive spelling correction at web scale. In *ICMLA*.

Martin Chodorow, Joel Tetreault, and Na-Rae Han. 2007. Detection of grammatical errors involving prepositions. In *Proceedings of the Fourth ACL-SIGSEM Workshop on Prepositions*, pages 25–30.

Silviu Cucerzan and David Yarowsky. 2002. Augmented mixture models for lexical disambigua-tion. In *EMNLP*.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*, Genoa, Italy.

Scott Deerwester, Susan Dumais, George Furmas, Thomas Landauer, and Richar Harshman. 1990. Indexing by latent semantic analysis. *The American Society for Information Science*.

Dmitriy Dligach and Martha Palmer. 2008. Novel semantic features for verb sense disambiguation. In *ACL*.

Philip Edmonds. 1997. Choosing the word most typical in context using a lexical co-occurrence network. In *EACL*.

Mohammed Ali Elmi and Martha Evans. 1998. Spelling correction using context. In *COLING*.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. Liblinear: A library for large linear classification. *Machine Learning Research*, 9(1871-1874).

Rachele De Felice and Stephen G. Pulman. 2008. A classifier-based approach to preposition and determiner error correction in L2 English. In *Proceedings of COLING*, Manchester, UK.

Rachele De Felice and Stephen G. Pulman. 2009. Automatic detection of preposition errors in learner writing. *CALICO Journal*, 26(3).

Michael Gamon, Jianfeng Gao, Chris Brockett, Alex Klementiev, William B. Dolan, Dmitriy Belenko, and Lucy Vanderwende. 2008. Using contextual speller techniques and language modeling for ESL error correction. In *Proceedings of the International Joint Conference on Natural Language Processing (IJCNLP)*, pages 449–456, Hyderabad, India.

Andrew Golding and Dan Roth. 1996. Applying Winnow to context-sensitive spelling correction. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 182–190.

Andrew Golding and Dan Roth. 1999. A winnow-based approach to context-sensitive spelling correction. *Machine Learning*, 34(1-3):107–130.

Andrew Golding. 1995. A Bayesian hybrid method for context sensitive spelling correction. In *Proceedings* *of the Third Workshop on Very Large Corpora (WVLC-3)*, pages 39–53.

Na-Rae Han, Martin Chodorow, and Claudia Leacock. 2006. Detecting errors in English article usage by non-native speakers. *Natural Language Engineering*, 12(2):115–129.

Matthieu Hermet, Alain Désilets, and Stan Szpakowicz. 2008. Using the web as a linguistic resource to automatically correct lexico-syntactic errors. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC)*, pages 390–396, Marrekech, Morocco.

Emi Izumi, Kiyotaka Uchimoto, Toyomi Saiga, Thepchai Supnithi, and Hitoshi Isahara. 2003. Automatic error detection in the Japanese learners' English spoken data. In *Companion Volume to the Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 145–148.

Michael Jones and James Martin. 1997. Contextual spelling correction using latent semantic analysis. In *ANLC*.

Thomas Landauer, Darrell Laham, and Peter Foltz. 1998. Learning human-like knowledge by singular value decomposition: A progress report. *Advances in Neural Information Processing Systems*, 10:45–51.

Mirella Lapata and Frank Keller. 2005. Web-based models for natural language processing. *ACM Transactions on Speech and Language Processing*, 21:1–31.

John Lee and Ola Knutsson. 2008. The role of pp attachment in preposition generation. In *CICLING*.

Lidia Mangu and Eric Brill. 1997. Automatic rule acquisition for spelling correction. In *ICML*.

Saif Mohammad and Graeme Hist. 2006. Distributional measures of concept distance: A task-oriented evaluation. In *EMNLP*.

Alla Rozovskaya and Dan Roth. 2010. Training paradigms for correcting errors in grammar and usage. In *ACL*.

Magnus Sahlgren. 2006. *The Word-Space Model: using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces*. Ph.D. thesis.

Joel Tetreault and Martin Chodorow. 2008. The ups and downs of prepostion error detection in esl writing. In *COLING*.

Joel Tetreault, Jennifer Foster, and Martin Chodorow. 2010. Using parse features for preposition selection and error detection. In *ACL*.

Casey Whitelaw, Ben Hutchinson, Grace Y. Chung, and Gerard Ellis. 2009. Using the web for language independent spellchecking and autocorrection. In *ACL*.