Natural Language Engineering **1** (1): 000–000. Printed in the United Kingdom

Predicting Word Choice in Affective Text

M. GARDINER, M. DRAS
Macquarie University, North Ryde NSW 2109, Australia

(*Received 10 March 2015*)

Abstract

Choosing the best word or phrase for a given context from among candidate near-synonyms, such as *slim* and *skinny*, is a difficult language generation problem. In this paper we describe approaches to solving an instance of this problem, the lexical gap problem, with a particular focus on affect and subjectivity; to do this we draw upon techniques from the sentiment and subjectivity analysis fields.

We present a supervised approach to this problem, initially with a unigram model that solidly outperforms the baseline, with a 6.8% increase in accuracy. The results to some extent confirm those from related problems, where feature presence outperforms feature frequency, and immediate context features generally outperform wider context features. However, this latter is somewhat surprisingly not always the case, and not necessarily where intuition might first suggest; and an analysis of where document-level models are in some cases better suggested that, in our corpus, broader features related to the 'tone' of the document could be useful, including document sentiment, document author, and a distance metric for weighting the wider lexical context of the gap itself. From these, our best model has a 10.1% increase in accuracy, corresponding to a 38% reduction in errors. Moreover, our models do not just improve accuracy on affective word choice, but on non-affective word choice also.

1 Introduction

Choosing the best word or phrase for a given context from among candidate near-synonyms is a difficult language generation problem. While Natural Language Generation (NLG) systems do have to consider lexical choice in general, it is typically not at the level of the fine distinctions embodied in near-synonyms. These can be important, as noted by Inkpen and Hirst (2006), who describe an NLG system that incorporates a knowledge base of near-synonym differences: for example, the choice between the flattering *slim* and the less complimentary *skinny* is significant. Extensions of this general idea of focussing on near-synonyms include systems that can rewrite text to be more positive or more negative (Inkpen, Feiguina, & Hirst, 2006) — for example, as an email filter to soften a potentially inflamatory first draft, or as an intensifier for an opinion piece intended to persuade — or an intelligent thesaurus that can help a writer by ordering proposed word choices by the suitability of the context (Inkpen, 2007a). In the language generation context, it is clear that a full-featured system would need to handle the problem of choosing

among near-synonyms or paraphrases in order to select the most felicitous phrasing and at times to avoid serious errors in language production.

The first task defined to explore this problem is the near-synonym lexical gap problem formalised by Edmonds (1997),[1] more recently termed the Fill in the Blanks (FITB) task, in which a word is removed from a sentence of text, and a system is offered both the original word and some of its near-synonyms as potential replacements. Edmonds offers this example of the task, in which the system must choose which of *error*, *mistake* or *oversight* fits into the gap in this sentence:

(1) However, such a move also would run the risk of cutting deeply into U.S. economic growth, which is why some economists think it would be a big _____.

Edmonds used an unsupervised approach, testing it on seven sets of synonyms (synsets) from WordNet (Fellbaum, 1998). This work was extended by Inkpen (2007b), who used much more training data and a language model in an alternative unsupervised approach, which had much better results on Edmonds's test set. Gardiner and Dras (Gardiner & Dras, 2007a, 2007b), looking at a larger test set (58 synsets versus 7), found that affective near-synonyms (for example *bad*, *insecure*, *risky*...) and non-affective near-synonyms behave differently, and suggested that near-synonyms with affective differences might be amenable to different techniques from those that are effective for non-affective near-synonyms.

In this paper we explore this suggestion, drawing on work in sentiment analysis. First, we investigate a supervised approach; as in the first exploration of supervised methods in document sentiment classification, by Pang, Lee, and Vaithyanathan (2002), we start with a simple unigram model. Second, we then look at broader aspects of the document to use as features. We hypothesise that affective differences between near-synonyms, such as the difference in attitude between *slim* and *skinny*, may be more influenced by more general aspects of the document such as affect, than are near-synonyms that differ in other aspects. We test this hypothesis by applying proven techniques from the domain of sentiment analysis (Pang et al., 2002) to the FITB problem, while investigating some new feature types for this problem combined with feature weighting techniques.

In Section 2 we discuss some related work on lexical replacement and (briefly) on sentiment analysis; in Section 3 we describe our data; and in Section 4 we discuss the selection of appropriate baselines. In Section 5 we describe our experimental setup, followed by the definition of our unigram models. We discuss the results of these, which motivates the selection of some further features. Section 6 then describes further experiments based on the document-level features arising from Section 5, while Section 7 describes those based on a notion of weighting in the feature space.

[1] We note that the more general task of choosing the appropriate word for a gap based on word association measures dates back to Church, Gale, Hanks, and Hindle (1989) and Church and Hanks (1991).

2 Related Work

While lexical selection is an issue in several applications that involve computational production of language — for example, Machine Translation (MT) — we only focus on the work that has been motivated by the task of Natural Language Generation (NLG) (Reiter & Dale, 2000). In MT, for example, any choosing among near-synonyms that occurs is an implicit part of the overall statistical model;[2] in the NLG-motivated work we look at, choosing among near-synonyms has been framed as an explicit problem, the FITB task, which we discuss below.

2.1 Lexical Selection in NLG

The FITB task has been directly addressed by several authors. It was introduced by Edmonds (1997), who experimented with seven sets of near-synonyms — for example, one set consists of the words *responsibility*, *burden*, *obligation*, and *commitment* — and describes a statistical system which can predict the original choice of word[3] 55.7% of the time depending on a chosen set of near-synonyms, according to the aggregate figures of Inkpen (2007b). Edmonds (1999) reports an extension to all WordNet synsets, in which the precision is 74.5% against a most-frequent baseline of 73.3%. Inkpen (2007b), taking a supervised approach using Pointwise Mutual Information (PMI) scores of the left and right context and the presence of frequent words in the context, reports an ability to predict the author's original choice of word of about 66% overall on Edmonds' original seven hand-chosen test cases, an improvement over the baseline of about 20%.

Following these, several alternative approaches have been proposed. Islam and Inkpen (2010) used a language model built using data from the Google Web1T corpus (Brants & Franz, 2006). Wang and Hirst (2010) used Latent Semantic Analysis (Landauer & Dumais, 1997) — drawing on the success of Rapp (2008) on a related task, that of identifying potential synonyms rather than choosing the usage of them — to perform dimensionality reduction on the feature space, and compare document representations with context representations. Yu, Shih, Lai, Yeh, and Wu (2010) use feature values derived from the Google Web1T corpus as suggested in Gardiner and Dras (2007b), weighting feature values by the strength of the collocation with given near-synonyms. Islam (2011) proposed a two-phase method, categorising n-grams based on the position of the candidate word to fill in the blank within the n-gram, and defining a normalised frequency to rank candidates.

[2] There are some exceptions, such as the CBMT approach of Carbonell, Klein, Miller, Steinbaum, Grassiany, and Frei (2006), who explicitly generate near-synonyms in cases of poor decoder performance. They note: "To our knowledge, no other translation engine utilizes dynamically-generated word or phrasal synonymy to optimize translation results".

[3] In all of the work that follows, the evaluation has been to predict the original choice of word. There is always the possibility that an alternative will be acceptable, but this would require extensive human evaluation. In this article, we follow the standard evaluation.

All of these approaches were evaluated on the Edmonds set of seven test cases; all produced higher results, with the last of these, Islam (2011), reaching 75.4%.[4]

The FITB task is closely related to the lexical substitution task addressed at SemEval-2007 (McCarthy & Navigli, 2007). In this task, instead of being presented with a gap and being required to correctly predict the author's original word choice from among several alternatives, the systems are given the author's original word choice and asked to predict a suitable alternative; the gold standard is a set of alternatives that came from a human annotation exercise carried out by the task organisers. Further, unlike in the FITB task, the set of alternative words is not fixed. The results of this harder task are not directly comparable to the FITB task, and the ability of systems to predict the very best alternative word had precision and recall of not greater than 13%. Baseline systems derived from walking the WordNet hierarchy around a target word performed at about 10% precision and recall whereas a distributional similarity baseline performed at under 9% precision and recall.

The methods used by the two best performing lexical substitution systems on the task requiring the best single substitute, as ranked by McCarthy and Navigli (2007), were: a language model trained on the 10^{12} words of the Web1T 5-gram dataset (Brants & Franz, 2006) selecting from words drawn from all WordNet synsets of the target word and all neighbouring synsets (Yuret, 2007); and a voting system between several selection methods including a language model, a machine translation test and a Latent Semantic Analysis measure selecting words drawn from several sources including WordNet and Microsoft Encarta (Hassan, Csomai, Banea, Sinha, & Mihalcea, 2007). The major explanation for the considerably lower performance reported by the best systems on the lexical substitution task as opposed to the FITB task is presumably the number of substitutes that the lexical substitution systems consider, none of which is guaranteed to be the correct replacement. In the FITB task, typically systems are considering between 3 and 7 possible substitutes, and the correct replacement is guaranteed to be among them. A very recent investigation of various approaches to the lexical substitution task with multiple lexical resources can be found in Sinha and Mihalcea (2014).

2.2 Sentiment and Subjectivity Analysis

Sentiment analysis and the closely related field of subjectivity analysis is a field concerned with predicting the sentiment features of text. There are several major tasks: determining the overall opinion of a text such as a review and identifying and classifying parts of the document with their own sentiment; identifying subjective elements within a piece of text; identifying differing opinions within a text in, for example, a review with multiple parts; and identifying features within documents that contribute to overall subjectivity or sentiment classification.

[4] The indicative figures here are macroaverages. Reported microaverages are typically lower: 61.7% for Inkpen (2007b) and 70.8% for Islam (2011).

In the task of determining the overall opinion, the goal is to predict, or, in evaluating, re-predict a sentiment label assigned to a text. For example, a movie review might rate a movie as 9 out of a possible 10. The task then is, given the text of the review, to re-assign the correct label. The label may be fine-grained (correctly predicting a score of 9 out of 10) or coarse-grained (correctly predicting whether a review is positive or negative). Examples of approaches to this task include the unsupervised method of Turney (2002) inferring the individual sentiment values of words and using these to predict the sentiment of movie reviews; the exploration by Pang et al. (2002) of various machine learning approaches to the problem of classifying movie reviews; and the application by Gamon (2004) of machine learning techniques to a domain more difficult than movie reviews, customer feedback data. As an extension of this problem, some authors have examined determining the sentiment of different aspects of a document. For example, Snyder and Barzilay (2007) examine the ranking of different aspects of a restaurant review, such as food and ambience.

A partially separate strand of investigation is subjectivity analysis, fundamentally although not exclusively concerned with distinguishing subjective text expressing the private viewpoint of an actor from objective text. It may be concerned with identifying subjective parts of a single document as in Hatzivassiloglou and Wiebe (2000), Wiebe, Wilson, Bruce, Bell, and Martin (2004), or, as Pang and Lee (2008) explain, as a variety of genre classification into the genres of "editorial" and "news", giving examples such as Yu and Hatzivassiloglou (2003) and many following.

The two strands of research often make use of similar features, or each others' results. For example, Pang and Lee (2004) found that identifying subjective portions of a document could result in more accurate classifications of the overall sentiment of the original review.

3 Affective Text: Near-Synonyms and Corpora

As noted in Section 1, our focus is on the FITB task, and the characteristics of the candidate near-synonyms that might fill the gap. Various typologies for near-synonyms, in terms of the differences in nuance that they can embody, have been proposed; following are the distinctions made by, and terminology used by, three such typologies:

- semantic or denotational variation (*mist* and *fog*) and stylistic or connotational variation (*stingy* and *frugal*) (DiMarco, Hirst, & Stede, 1993);
- collocational and syntactic variations (*die* and *pass away*), stylistic variations (*house* and *habitation*), expressive variations (*skinny* and *slim*) and denotational variations (*error*, *blunder* and *mistake*) (Edmonds & Hirst, 2002); and
- denotational differences (*invasion* and *incursion*), attitudinal variations (*placid* and *unimaginative*) and stylistic differences (*assistant* and *helper*) (Edmonds & Hirst, 2002; Inkpen & Hirst, 2006).

We are just interested in a broader classification, into either affective or non-affective, as in our earlier work of Gardiner and Dras (2007a, 2007b): whether the near-synonyms differ in sentiment expressed towards their subject, or whether they differ in some other way.

In this paper we explore the idea that affective and non-affective synonyms might behave differently in the context of the FITB task. To do this, we require both a set of near-synonyms and a set of documents with known document sentiment. Our choice of these two datasets is described in the following subsections.

3.1 Documents containing sentiment

As samples of our test words in context, and a standard source of affective text, we took the widely used Movie Review Corpus sentiment scale data set, version 1.0 (SCALE 1.0) introduced by Pang and Lee (2005). The data consists of movie reviews authored by four reviewers on Internet sites, ranging from extremely negative to extremely positive reviews (there are 5000 short reviews in total, average length around 380 words). Scores are given on a 10-point scale, normalised to the range [0,1]; the label 'positive' in this data is assigned to scores ≥ 0.7, and the label 'negative' is assigned to scores ≤ 0.4. This score is available as the RATING attribute of each review.[5]

3.2 Sentiment annotated near-synonyms

Since we seek to examine the choice between near-synonyms that differ in sentiment we require a source of such near-synonyms. There are several sources of near-synonyms not annotated for polarity: the major one used in previous work was WordNet (Fellbaum, 1998) used by Edmonds (1997), Inkpen (2007b) and Gardiner and Dras (Gardiner & Dras, 2007a, 2007b) to test their near-synonym usage prediction methods. However not only is WordNet not annotated for polarity, it encodes very fine-grained sense distinctions which usually precludes having near-synonyms that differ in sentiment contained in a single synset. In addition, there were very few synsets — only 7 of the 58 used in the Gardiner and Dras work — that had any affective meaning, and as we are especially interested in these sets, we require an additional source of near synonyms.[6]

There are some versions of WordNet annotated for sentiment, for example SentiWordNet (Esuli & Sebastiani, 2006), but examination of this data shows that it is not easy to use it to produce clear distinctions such as "this set of near-synonyms differ in sentiment and these do not". As an example of why SentiWordNet is a difficult source of data for this use-case, consider the five entries from it shown

[5] `http://www.cs.cornell.edu/people/pabo/movie-review-data/scaledata.README.1.0.txt`

[6] As an illustration of why WordNet synsets groupings aren't quite what we'd want, we note that our initial examples of *slim* and *skinny* aren't in the same synset: *slim* is in a synset with *slender*, *slight*, and *svelte*; *skinny* is in a synset with *scraggy*, *scraggly*, *boney*, *bony*, *scrawny*, *skinny*, *underweight*, *weedy*.

POS	Words	Positive score	Negative score
Adjective	*rich*, *plentiful*, *plenteous*, *copious*, *ample*	0·125	0
Verb	*merit*, *deserve*	0·75	0·125
Verb	*swell*, *puff up*	0·125	0·625
Noun	*feat*, *exploit*, *effort*	0·375	0·125
Noun	*swan song*, *last hurrah*	0·125	0·125

Table 1. *Example entries from SentiWordNet*

in Table 1. While having real-numbered values for the positivity and negativity of synsets would be of use if we were seeking features for a sentiment classification learner, one of the uses to which SentiWordNet has been put, but it is not straightforwardly apparent how to identify "'synsets that have the same affect", "synsets that differ in affect" and "synsets that have no affective meaning" from the numerical values.

There are other resources such as the General Inquirer word list (Stone, Dunphy, Smith, & Ogilvie, 1966) which have been human-annotated with sentiment, but which are not already grouped into near-synonym sets, and are only concerned with polarity, rather than degrees of sentiment.

The near-synonym usage guide *Choose the Right Word* (Hayakawa, 1994) contains near-synonym sets chosen by a human author specifically as a guide to the subtleties of near-synonym word choice for readers and writers of English. It is therefore a good source of near-synonyms that differ in fundamental ways such as sentiment. It was used by Inkpen and Hirst (2006) as a source of near-synonyms marked for differences such as denotational differences (in which near-synonyms could differ in what they *suggest* or *imply*, for example) and attitude and style differences, including near-synonyms that are more *pejorative*, *disapproving* or *favourable*. In addition, unlike resources such as SentiWordNet, it has explicit characterisations of the differences between the near-synonyms.

Inkpen and Hirst (2006) derived data automatically from *Choose the Right Word* by a decision list algorithm, and this data is not available for reasons of copyright. In addition, the focus of this data was on all axes in which near-synonyms can differ, rather than on the axis of positive or negative attitude to the subject of a description in particular. Thus we have annotated our own data.

We use sets of near-synonyms drawn from an earlier edition of this work, *Use the Right Word* (Hayakawa, 1968). (An excerpt from an entry in *Use the Right Word* is shown in Figure 1 as an example of its contents.[7]) As a first pass, we sampled the

[7] By contrast, SentiWordNet has self-contained entries for individual words — for example, *asperse* has the definition "charge falsely or with malicious intent; attack the good name and reputation of someone; 'The journalists have defamed me!' 'The article in the paper sullied my reputation'." `http://sentiwordnet.isti.cnr.it/search.php?q=asperse`

malign, *asperse*, *defame*, *libel*, *slander*, *vilify*
These words mean to say or write something, often misleading or false, that is damaging to a person or a group of people. *Malign* is perhaps the broadest word in the group in that the feelings which motivate a person who *maligns* another can range from ... simple ill will ... to bitter hatred ...
Asperse and *vilify* imply false accusations made in order to ruin someone's reputation ... *Apserse* however is extremely formal, and more commonly appears in the form of a plural noun ...
Defame can specifically indicate an attempt to destroy someone's good name ...
Libel and *slander*, in their most restricted sense, are legal terms pertaining to defamation ... In popular usage, however, both words are applied to false accusations by any means. See *accuse*, *belittle*, *lie*.
ANTONYMS: *praise*

Fig. 1. Excerpt from an entry in *Use the Right Word*

sets by annotating those that are listed under the letter A, of which there are 57 in total. Of these 57 total sets, we exclude 16 sets that do not include at least two words that are each used at least five times in our sentiment annotated documents described in Section 3.1. This left 37 sets, totalling 133 words. Because this dataset only contained 5 sets with the same affect shared among all words (see annotation description immediately following), we added an extra 10 sets (also from *Use the Right Word*, but beyond the letter A) with the same affect shared among all the words to balance the number of test sets.

We thus have a total of 47 test sets. These near synonyms were annotated for affect by one of the authors. The annotation scheme called on the annotator to rely on *Use the Right Word*'s interpretation of the sets, rather than personal linguistic intuition. For example, *Use the Right Word* suggests in the entry for *aloof*: "Both *reserved* and *detached* can be associated with attractive qualities, whereas *aloof* is seldom so considered". Annotations were of two kinds:

1. for a given set, whether *Use the Right Word* indicates that that set contains at least some words conveying sentiment ('affective', or 'not affective'); or
2. for every word within any set marked as 'affective', whether *Use the Right Word* indicates that that word has positive, negative, or neutral affect.

Near-synonym sets were then given one of three labels: NONE, for those where no element of the set conveyed sentiment; SAME-AFFECT, for those which contain only positive or only negative words (not neutral ones); or DIFFERING-AFFECT, where the words contained different sentiment. (Note that DIFFERING-AFFECT sets can contain neutral words, such as the set consisting of positive *insight* and neutral *perception*.)

An example of three annotated test sets is shown in Table 2. Observe that the set containing *attend* and *accompany* has set type marked as NONE, meaning no affect. The marking of Neutral against the individual words is thus implied. The set containing *ludicrous*, *senseless* and others is marked Same, and all the words

Set sentiment type	Word	Sentiment	Word	Sentiment	Word	Sentiment
Same	*ludicrous*	Negative	*senseless*	Negative	*foolish*	Negative
	preposterous	Negative	*ridiculous*	Negative	*farcical*	Negative
	absurd	Negative	*silly*	Negative	*irrational*	Negative
	unreasonable	Negative				
None	*attend*	Neutral	*accompany*	Neutral		
Differing	*precise*	Neutral	*accurate*	Neutral	*exact*	Neutral
	right	Positive	*nice*	Neutral	*correct*	Neutral
	true	Neutral				

Table 2. *Examples of test sets annotated for overall set sentiment differences, and for the sentiment of individual words*

are indeed marked identically with Negative sentiment. The set containing *precise*, *accurate* and others is marked Differing and the sentiment of the individual words does indeed differ, with words within it having varying Positive and Neutral sentiment. The complete set of annotated near synonym sets is given in Appendix A.

We divided the sets into development and test data; examples were inspected from the development data during development of the models. The distribution of near synonym sets is then shown in Table 3. The number of instances used in our evaluation (development set: 12656; test set: 16118) is comparable to the classic Reuters-21578 corpus used in text classification (21578 instances) or the development and test sets used in the 2013 Workshop on Machine Translation shared task.[8]

4 Comparison of baselines

There are a number of possible baselines, each with various merits. As candidates in this paper, we consider two standard approaches (most frequent category, and a language model-based one); and two based on methods published in the literature, those of Edmonds (1997) and Inkpen (2007b), for which we use the implementations described in Gardiner and Dras (2007a) and Gardiner and Dras (2007b) respectively. We will here refer to those implementations as EDMONDS-COLLOCATE and WEB1T-PMI.

[8] `http://www.statmt.org/wmt13/translation-task.html`

Set type	Development set					Test set				
	No.	Mean	Min	Max	Inst.	No.	Mean	Min	Max	Inst.
All sets	12	3.6	2	6	12656	35	3.6	2	10	16118
NO-AFFECT words	8	3.1	2	5	8587	12	2.8	2	4	7303
SAME-AFFECT words	2	4.5	4	5	310	13	3.5	2	10	4441
DIFFERING-AFFECT words	2	4.5	3	6	3759	10	3.9	2	7	4374

Table 3. *Distribution of sentiment among Use the Right Word sets in* SCALE 1.0 *corpus: number of sets with a particular sentiment; mean, minimum and maximum number of words in sets; number of instances.*

We outline these below, and discuss their relative performance, along with the resulting choice of baseline for the rest of the paper.

4.1 Candidate baselines

4.1.1 Most frequent

With respect to baselines, Edmonds (1997) and Inkpen (2007b) both use a most frequent baseline, that is, comparing with the method of always selecting the most frequent word in a set to fill the gap. This baseline can be quite high: Inkpen reports it as achieving 44.8% accuracy on the seven sets of test words used by herself and Edmonds (these seven sets contain between two and four words, with a mean of exactly three words per set). However, we find that on our dataset, even the most-frequent baseline is considerably higher for most sets in our development set, as shown in Table 4.

4.1.2 Language model

As Inkpen (2007b) notes, and as seen in the 2007 lexical substitution task, statistical language models are the mainstream method of lexical choice. Inkpen and Hirst (2006) compared their system to a language model baseline that was implemented as part of the HALogen NLG system (Langkilde & Knight, 1998), trained on 250 million works of text from the news genre. HALogen's word choices when combined with the anti-collocation method presented by Inkpen and Hirst (2006) outperformed HALogen alone, and the method presented by Inkpen (2007b) outperforms anti-collocations, and thus Inkpen (2007b) concludes that language models would be outperformed by her newer method.

Here we implement a baseline language model choice system using the Web1T data (Brants & Franz, 2006). Web 1T contains n-gram frequency counts, up to and including 5-grams, as they occur in a trillion words of World Wide Web text. There is no context information beyond the n-gram boundaries. Examples of a 3-gram

and a 5-gram and their respective counts from Web 1T are shown in examples (2) and (3):

(2) `means official and 41`

(3) `Valley National Park 1948 Art 51`

We make a word choice by estimating the most probable 3-gram,[9] backing off to 2-grams and 1-grams where necessary. We fundamentally use the backoff method of Katz (1987). The standard approach to Katz uses smoothing below some threshold to avoid data sparseness for low-frequency items, with this threshold often empirically determined, and uses the more accurate raw high-frequency accounts above it. Given that Web1T only provides counts frequencies of at least 40 for bi- to 5-grams and 200 for unigrams, we use these as our threshold. Full details of our adaptation of Katz to the Web1T data are given in Appendix B.

4.1.3 EDMONDS-COLLOCATE *prediction method*

Here we describe our implementation of Edmonds (1997). For the gap in sentence S, each candidate token c — where a token is a part-of-speech tagged word, such as *(JJ arduous)* or *(NN fight)* — is assigned a score, $score(c, S)$, which is the sum of its significance score with each individual remaining token w in that sentence:

$$\text{score}(c, S) = \sum_{w \in S} \text{sig}(c, w) \tag{4}$$

The candidate c which maximises $score(c, S)$ is chosen as the word fitting the lexical gap in sentence S.

Edmonds computed the score $sig(c, w)$ by connecting words in a COLLOCATION NETWORK. The principle is that if word w_0 co-occurs significantly with word w_1 which in turn co-occurs significantly with word w_2, then the presence of w_0 should weakly predict the appearance of w_2 even if they do not significantly co-occur in the training corpus. That is, he assumes that if, for example, *task* co-occurs significantly with *difficult*, and *difficult* co-occurs significantly with *learn*, then *task* and *learn* should weakly predict each other's presence.

Edmonds proposes extending this technique to co-occurrence networks with prediction chains of arbitrary length, but his experimental results suggest that in practice two connections approaches the limit of the usefulness of the technique. Therefore, to compute $sig(c, w)$ we take the shortest path of significance between the tokens c and w, which is either c, w where c and w significantly co-occur, or c, w_0, w where c and w both significantly co-occur with a third word, w_0.

Where tokens c and w significantly co-occur together, their significance score is their t-score (Church, Gale, Hanks, & Hindle, 1991), which we calculate by the Ngram Statistics Package (Banerjee & Pedersen, 2003):

[9] Specifically, say we had candidates c_1, c_2 and c_3 for filling the gap. We consider the 3-gram w_1 *GAP* w_2, and look for the most probable of $w_1c_1w_2$, $w_1c_2w_2$ and $w_1c_3w_2$ to choose between c_1, c_2 and c_3.

(5) $$\text{sig}(c, w) = t(c, w)$$

The t-score is calculated by comparing the likelihood of both words c and w occurring within a certain window of each other. The size of the window is either a 4 word window surrounding c, that is, c and w were found at most 2 words apart, or a 10 word window surrounding c, that is, c and w were found at most 5 words apart.

Where tokens c and w both significantly co-occur with token w_0, their significance score is a combination of their t-scores, with a bias factor devised by Edmonds to account for their weaker connection.

(6) $$\text{sig}(c, w) = \frac{1}{8}(t(c, w_0) + \frac{t(w_0, w)}{2})$$

If there is more than one candidate word w_0 co-occurring significantly with both c and w, the word w_0 is chosen so that the value of $\text{sig}(c, w)$ in equation (6) is maximised.

Where more than one candidate word c maximises $\text{sig}(c, w)$ or where $\text{sig}(c, w) = 0$ for all candidate words c, we back off to the baseline result of the most frequent word.

In the above, we have used "significantly co-occur" without definition. The test we are using is that from the description by Edmonds (1999) of the same experiment: any two words w_0 and w_1 significantly co-occur if their t-scores are greater than 2.0 and their mutual information score is greater than 3.0, as suggested by the observation of Church et al. (1991) that t-scores and mutual information scores emphasise different kinds of co-occurrence.

Input to the t-score and mutual information systems was the part-of-speech tagged 1989 *Wall Street Journal*, as in Edmonds's work. Stop words were those used by Edmonds, defined as any token with a raw frequency of over 800 in the corpus, and all punctuation, numbers, symbols and proper nouns. Per Edmonds we did not perform lemmatisation or word sense disambiguation.

4.1.4 WEB1T-PMI *prediction method*

In the near-synonym prediction method described by Inkpen (2007b), here called INKPEN-PMI, the suitability of candidate c for a given gap is approximated differently from EDMONDS-COLLOCATE: the entire sentence is not used to measure the suitability of the word. Instead, a certain sized window of k words either side of the gap is used. For example, if $k = 3$, the word missing from the sentence in example (7) is predicted using only the six words shown in example (8).

(7) Visitors to Istanbul often sense a second, _____ layer beneath the city's tangible beauty.

(8) sense a second, _____ layer beneath the

Given a text fragment f consisting of $2k$ words, k words either side of a gap g $(w_1, w_2, \ldots, w_k, g, w_{k+1}, \ldots, w_{2k})$, the suitability $s(c, g)$ of any given candidate word c to fill the gap g is given by:

$$s(c, g) = \sum_{j=1}^{k} PMI(c, w_j) + \sum_{j=k+1}^{2k} PMI(w_j, c) \tag{9}$$

INKPEN-PMI estimates the token counts for $PMI(x, y)$ by issuing queries to the Waterloo MultiText System (Clarke & Terra, 2003) for occurences of x and y separately and within a *query frame* of length q within a corpus.

According to Inkpen (2007b), INKPEN-PMI outperformed both the baseline and EDMONDS-COLLOCATE by 22 and 10 percentage points respectively on the seven synsets from Edmonds (1997).

In this work, without access to the Waterloo Multitext System and the corpus used by Inkpen, we require an approximation to the method. WEB1T-PMI, our variation on INKPEN-PMI, is designed to estimate $PMI(x, y)$, the pointwise mutual information of words x and y, using the Web 1T 5-gram corpus Version 1 (Brants & Franz, 2006) (described in Section 4.1.2).

The n-gram counts of Web1T allow us to estimate the occurence of x and y within a query frame k by summing the Web 1T counts of k-grams in which words x and y occur and x is followed by y. Counts are computed using the Web 1T processing software "Get 1T" detailed in Hawker, Gardiner, and Bennetts (2007). Queries are matched case-insensitively, but no stemming takes place, and there is no deeper analysis (such as part of speech matching).

This gives us the following methodology for a given lexical gap g and a window of k words either side of the gap:

1. for every candidate near-synonym c:
   (a) for every word w_i in the set of words preceding the gap, $w_1, \ldots, w_k$, calculate $PMI(w_i, c)$, given counts for occurences of w_i, c and w_i and c within a query frame from Web 1T;[10]
   (b) for every word w_j in the set of words following the gap, $w_{k+1}, \ldots, w_{2k}$, calculate $PMI(c, w_j)$ as above;
   (c) compute the suitability score $s(c, g)$ of candidate c as given by equation (9);
2. select the candidate near-synonym with the highest suitability score for the gap where a single such candidate exists;
3. where there is no single candidate with a highest suitability score, select the most frequent candidate for the gap (that is, fall back to the baseline described in Section 4.1.1).[11]

Since Web 1T contains 5-gram counts, we can use query frame sizes from $q = 1$ (words x and y must be adjacent, that is, occur in the 2-gram counts) to $q = 4$.

[10] Where the counts are 0 for the purpose of computing $s(c, g)$, we define $PMI(x, y) = 0$ so that it has no influence on the score $s(c, g)$ given by equation (9).
[11] Typically, in this case, all candidates have scored 0.

4.2 Results and discussion

We discuss each individual potential baseline with respect to the most frequent baseline below. In summary, all performed worse, so we use the most frequent baseline in the rest of the paper. Comparative results of all candidate baselines are shown in Table 4.[12, 13]

4.2.1 Language model results

The language model badly underperformed compared to the most frequent baseline, which was not expected from the results of Inkpen and Hirst (2006). It is, however, difficult to compare this result directly with the one given in Inkpen and Hirst (2006). In that article, the HALogen system's language model, which predicts the correct near-synonym between 58% and 83% of the time, is given as a baseline against which to compare the Xenon experimental system, but values for the most frequent baseline are not given for the same test sets.

One possible reason for the bad performance of the language model is that the most frequent baseline here, as noted earlier, is substantially higher than in other work; perhaps there is something about the nature of our near-synonym sets. A second possible reason that could interact with the first is suggested by Inkpen and Hirst (2006), who note that the collocations encoded in the language model will tend to be with function words, per the short n-gram distance. However in light of the good performance of language models on the FITB task described in Islam and Inkpen (2010), also trained on Web1T, this would be worth investigating further. There are some differences in our language model implementation to that of Islam and Inkpen, including a different smoothing technique and the use of 3-grams rather than 5-grams. However, without a high performing language model, we do not consider it as a baseline in this work.

4.2.2 Edmonds-collocate *results*

Table 4 includes results for Edmonds-collocate where the query window size is 4. Results for query window size 10 are generally very slightly worse again. In fact, the method only makes predictions for at most 10% of the time (and for some of the sets, makes no predictions at all): the back-off strategy accounts for its relative closeness to the most frequent baseline. This is unsurprising given the finding that Edmonds-collocate simply does not use a large enough training set to make useful predictions (Inkpen, 2007b).

[12] Results are reported as percentage points above or below the most frequent baseline. The language model accuracy, for example, is 72.3% - 18.1% = 54.2%.

[13] This was something of a surprise — in the published works, all performed better than the baseline used there (of Edmonds's seven synsets) — which could have suggested an error in implementation. However, we used the implementations of Edmonds-collocate and Web1T-PMI of Gardiner and Dras (2007b), which were tested against the Edmonds synsets and produced comparable results to the originally published ones, so it is unlikely to be that.

Potential baseline	Overall			No affect			Same affect			Differing affect		
	dev.	test	both	dev.	test	both	dev.	test	both	dev.	test	both
MF baseline	72·3	73·8	73·1	82·6	85·2	83·8	44·8	71·4	69·6	50·9	57·1	54·2
Language model	−18·1	−15·4	−16·8	−22·0	−21·9	−21·9	+1·0	−40·6	−35·8	−10·8	+0·5	−4·8
EDMONDS-COLLOCATE (query size = 4)	−2·8	0·0	−1·3	−3·7	0·5	−1·7	−3·2	−0·5	−0·6	−0·9	−0·5	−0·7
WEB1T-PMI ($q = 2$,$w = 2$)	−18·6	−18·6	−18·6	−22·3	−22·2	−22·3	+9·0	−31·3	−28·6	−12·3	+0·1	−5·6

Table 4. *Percentage point increase of alternative baselines over most frequent baseline*

4.2.3 WEB1T-PMI *results*

The best performance for WEB1T-PMI parameters was $q = 2$ and $w = 2$, which are shown in comparison with other baselines in Table 4. One noteworthy characteristic of the varying parameters is that a wider window (parameter k) of context around a gap almost always diminishes performance.

In general these results also show a large decrease in performance over the baseline, contrary to results reported in Gardiner and Dras (2007b), where results were approximately equal to the baseline. Several reasons may hold as to why WEB1T-PMI performs unexpectedly poorly. First, the movie review test data in this article is very different from that used in Gardiner and Dras (2007b), which tested on the Wall Street Journal; perhaps movie review text is quite different from the aggregate represented by Web1T. Second, the test near-synonyms in this article are also different from those used in previous work, being near-synonyms selected by a human editor rather than high frequency WordNet synsets. Third, the test near-synonyms in Gardiner and Dras (2007b) were trimmed to only three or four possible choices, whereas this set contains up to 9 possible alternatives.

5 Unigram models

In this section we examine the effectiveness of simply using unigrams and a machine learner, as Pang et al. (2002) did for document sentiment classification. We then examine their performance relative to our chosen baseline from Section 4, choosing the most frequent word.

The learner We use the SVM-Light implementation of Support Vector Machines (Joachims, 1999), which implements a binary classifier. Therefore a separate machine m_{c_i} is trained for each binary decision: is the gap filled by candidate word c_i or not? We select the SVM m_c from among the set that returns the highest confidence score (as suggested by Rifkin and Klautau (2004) and Liu and Zheng (2005)) and choose c to fill the gap. c is judged correct if it matches the word w the original author used.

Hypothesis testing A single method for comparing classification accuracy has not been universally accepted. What constitutes an invalid method is now more widely recognised: Salzberg (1997), for example, points out both the incorrectness of using a regular (two-sample) t-test on two accuracy scores, and the surprisingly widespread use of it in the machine learning community at that time. One circumstance when regular t-tests are incorrectly used is when the classifiers are evaluated on the same data set, so a test that requires independent data, as a t-test does, is inappropriate. He consequently defines an approach that uses a paired test — a paired t-test, McNemar test or similar — and k-fold cross-validation. The survey on cross-validation by Refaeilzadeh, Tang, and Liu (2009) notes that more complicated proposals, such as various $n \times k$-fold cross-validation approaches, have not yet been widely accepted. We therefore use a simple 5-fold cross-validation.

With respect to the statistical test, our data, with the relatively high baselines as noted in Section 4 but with some quite low values as well, appears to be quite skewed. The Gaussian-based t-test is therefore not suitable, and we use the non-parametric McNemar test (Sprent & Smeeton, 2007) instead.[14]

5.1 The models

To construct our unigram features, we consider every other word in the context of the gap as a feature used to predict the correct word for the gap, giving us a feature space equal to the number of distinct tokens in the corpus. In producing the features for each set of words, we excluded all words in the set being predicted. We also, following Pang et al. (2002), excluded all tokens that did not occur at least 4 times in the training data, and we did not use stemming or stop lists.[15]

We tested four possible unigram models:

1. the frequency of each word in the document containing the gap (DocFreq)
2. the presence of each word in the document containing the gap (DocPres)
3. the frequency of each word in the sentence containing the gap (SentFreq)
4. the presence of each word in the sentence containing the gap (SentPres)

5.2 Results and discussion

The increase in prediction accuracy for each of these four baselines over the most frequent baseline is shown in Table 5.

First, as seen in Gardiner and Dras (2007a) and Gardiner and Dras (2007b), we observe that the most frequent baseline itself differs for near-synonym sets with and without attitudinal meaning, and also when that meaning is the same or differs among the near synonyms. Observe in particular that the performance of the most frequent baseline is lower for attitudinal near-synonym sets, providing some support for suggesting that choosing between these may be a more difficult task.

Second, like Pang et al. (2002) on this same data set, we have found that presence features equal or outperform frequency features, even though we are performing a fairly different task, lexical gap prediction rather than classifying documents by sentiment. The improvement of presence over frequency is less dramatic at the sentence level than the document level, presumably because most tokens will only occur at most once in a sentence in any case.

Third, the improvement of only using tokens in the sentence surrounding the

[14] Foody (2008) gives a good overview of the use and applicability of these tests in the comparable field of machine learning for imaging.

[15] We had three reasons for not using stop lists, even though they are common in many classification tasks. In addition to following Pang et al. (2002), we considered that function words are useful in stylistic classification — see e.g. Koppel, Akiva, and Dagan (2006a) – and might capture some stylistic characteristics relevant to near-synonymy. Further, it made the extensions of Section 7 more straightforward.

Unigram model	Overall			No affect			Same affect			Differing affect		
	dev.	test	both	dev.	test	both	dev.	test	both	dev.	test	both
MF baseline	72·3	73·8	73·1	82·6	85·2	83·8	44·8	71·4	69·6	50·9	57·1	54·2
DocFreq	+3·2	+1·3	+2·1	+0·0	+1·5	+0·7	+10·0	+1·5	+2·1	+9·9	+0·7	+5·0
DocPres	+5·3‡	+2·0‡	+3·4‡	+0·2†	+2·1‡	+1·0‡	**+13·5‡**	**+2·4‡**	**+3·1‡**	+16·4‡	+1·4†	+8·4‡
SentFreq	+8·8	+4·6	+6·5	+2·5	+4·1	+3·3	+10·3	+1·5	+2·1	+23·1	+8·5	+15·3
SentPres	**+9·5$^{\ddagger\vee}$**	**+4·8$^{\ddagger\vee}$**	**+6·8$^{\ddagger\vee}$**	**+2·9$^{\ddagger\vee}$**	**+4·4$^{\ddagger\vee}$**	**+3·6$^{\ddagger\vee}$**	+11·6‡	+1·7‡	+2·3$^{\ddagger\times}$	**+24·2$^{\ddagger\vee}$**	**+8·7$^{\ddagger\vee}$**	**+15·9$^{\ddagger\vee}$**

Bold values are best performance for that column

* Difference from most frequent baseline significant at the $p < 0.05$ level (shown for DocPres and SentPres only)

† Difference from most frequent baseline significant at the $p < 0.01$ level (shown for DocPres and SentPres only)

‡ Difference from most frequent baseline significant at the $p < 0.001$ level (shown for DocPres and SentPres only)

$^{\times}$ Difference from DocPres significant at the $p < 0.05$ level (shown for SentPres only)

$^{\cup}$ Difference from DocPres significant at the $p < 0.01$ level (shown for SentPres only)

$^{\vee}$ Difference from DocPres significant at the $p < 0.001$ level (shown for SentPres only)

Table 5. *Percentage increase of unigram models over* most frequent *baseline*

gap over using tokens in the entire document in general echoes the result of Edmonds, that using features from a wide window (in his case, 50 words) around a gap diminished performance over using a smaller window. This is similar to other tasks such as word sense disambiguation (WSD): the only survey comprehensively discussing the issue of window size in WSD (Ide & Vronis, 1998) noted that the use of 'micro-context' provided most benefit, and the use of 'topical context' was variable and generally minimal.

Fourth, and most interestingly in that it is new and is relevant to our hypothesis, and not reflected in the literature to date, is that there is an exception to the second point, which is that SAME-AFFECT words are more accurately chosen using DOCUMENT FREQUENCY rather than SENTENCE FREQUENCY.

A possible explanation for this is that, whereas NO-AFFECT and different-affect have sufficient information at the sentence level, for SAME-AFFECT sets there is a 'tone' that suffuses the document that is important in replicating word choice. As an example from the Movie Review Corpus of this kind of tone distributed throughout a document (our italics): "even though the film *suffers* from its *aloof* and *uninviting* approach ... the *problem* with the picture seems on the surface to be its *plodding* pacing, but actually the *defect* has to do more with ...".

This is the sort of phenomenon that has been discussed by corpus linguists under the notion of semantic prosody or discourse prosody, first attributed in the literature to Sinclair (1987), who later noted that "many uses of words and phrases show a tendency to occur in a certain semantic environment" (Sinclair, 1991, p112), illustrating this by the generally negative nature of the grammatical subjects of the phrasal verb *set in* (e.g. *rot*, *malaise*, *disillusion*). Some researchers have preferred the term discourse prosody to emphasise the discourse-spanning nature of the phenomenon, with an emphasis on attitude (Stubbs, 2001, p65): "A discourse prosody is a feature which extends over more than one unit in a linear string .... Discourse prosodies express speaker attitude." An overview of the extensive corpus linguistic work on the topic is given in Stewart (2010).

Of the twelve sets of SAME-AFFECT words in the development and test set, with 4751 instances in the corpus, only the smallest set (with 35 instances) was of positive affect; the other eleven sets and 4716 instances were negative. That is, there was vastly more negative language, and the somewhat low baseline for these (69.6%) possibly suggests some variety in insults used to criticise the movies. This would fit with the work of Wiebe et al. (2004) on subjectivity, where they found that subjective opinion pieces exhibited a greater amount of linguistic 'creativity' — "Apparently, people are creative when they are being opinionated" — as evidenced by aspects such as higher frequency of *hapax legomena*. This then suggested to us two possible ways that this 'tone' might be manifest in the movie reviews: in the overall sentiment of the document, or through the writing style of a particular author.

That document sentiment might be useful is intuitive, and part of our reason for reviewing document sentiment analysis techniques in Section 2.2. The result that document-level classifiers did worse than sentence-level ones on DIFFERENT-AFFECT sets of words (as opposed to SAME-AFFECT ones) went against that intuition, and

	aggregate	uniform
author #1	0·032	0·121
author #2	0·086	0·234
author #3	0·041	0·088
author #4	0·010	0·180

Table 6. *Kullback-Leibler divergences for authors vs aggregate and uniform distributions*

led to the authorial style idea inspired by the work of Wiebe et al. (2004). To check this idea that a particular author's style might be distinguishable (and therefore perhaps useful in detecting this creative choice of SAME-AFFECT words), we did a quick analysis of the development set. With most of the SAME-AFFECT instances being negative, we looked at the distribution of negative words as broken down by the 10-point rating scale (from 0.1 to 1.0), for each author. We then compared each distribution against the aggregate distribution for all authors and against a uniform distribution as measured by their Kullback-Leibler divergence (Kullback & Leibler, 1951); results are in Table 6. Kullback-Leibler divergence gives a value for the difference between two distributions, or rather, the inadequacy of one distribution as a model for another.

Note that scores for individual authors vary by a factor of 4 (vs uniform) or 8 (vs aggregate). There is no generally agreed interpretation of absolute Kullback-Leibler divergence values, but the point to be drawn here is that some authors are much more different from the typical case in their use of negative words than are others. Also of note is that use of particular near synonyms and other similar linguistic phenomena where language allows a certain amount of choice at a local level is useful in the opposite task: given a document with certain features, identify its author (Koppel, Akiva, & Dagan, 2006b). Supporting the idea that author information is useful from a different type of source, author-topic models — topic models of documents, defined via a generative model, that include a model of the author — can display improved predictive power beyond document models of just topic alone (Rosen-Zvi, Griffiths, Steyvers, & Smyth, 2004).

We discuss our models incorporating document sentiment and author information in the next section.

6 Sentiment-derived features with unigrams

6.1 Document sentiment

We constructed two types of models to incorporate document sentiment. The first was just to use the sentiment of each review given in the movie review corpus. This is a gold-standard sentiment, but does not contribute many features. Our second type of model was to use the sentiment of individual words in the document, the

sort of feature often used in document sentiment classification; for this, we used scores from MicroWNOP (Cerini, Compagnoni, Demontis, Formentelli, & Gandi, 2007), a subset of WordNet annotated with polarity information.

MicroWNOP assigns scores to synsets based on their perceived Positive and Negative semantics as judged by human annotators. Cerini et al. (2007) divide the synsets into three groups:

- the *Common* group, with one Positive and one Negative score per synset (with the pool of five annotators working collaboratively);
- the *Group1* group, with three Positive and three Negative scores per synset (three annotators giving individual annotations); and
- the *Group2* group, with two Positive and two Negative scores per synset (the remaining two annotators giving individual annotations).

An example entry from *Group1* is *baseborn*, *humble*, and *lowly*, with all three annotators assigning it 0 for positivity; the first two assigned 0.5 for negativity, and the third 0.25 for negativity. As with SentiWordNet, the entire synset is assigned the score, not any individual words within it.

6.1.1 Gold-standard sentiment

The first set of features, DocSent, is as follows:

- the sentiment of the document in question, as assigned by the rating measurement[16] of the Scale 1.0 corpus; and
- the sentiment of the target word, as assigned in the annotation (Section 3.2).

6.1.2 Approximate sentiment

This set of features includes the features from Section 6.1.1 and adds features from MicroWNOP. We construct the following definition of a single MicroWNOP Positive score and MicroWNOP Negative score for a word. The MicroWNOP Positive score for a word is the highest single Positive score assigned to a synset containing that word, whether assigned in the Common, Group1 or Group2 categories. The MicroWNOP Negative score is the equivalent value for the negative scores.

We then define four features, intended to be a proxy for the document sentiment:

- the sentiment of the target word, as assigned by annotators in Section 3.2;
- the average of the MicroWNOP Positive scores of all of the words in the document, excepting the target word;
- the average of the MicroWNOP Negative scores of all the words in the document, excepting the target word; and
- the total MicroWNOP Positive scores of all the words in the document, excluding the target, minus the MicroWNOP Negative scores of all the words in the document, excluding the target.

[16] See Section 3.1.

6.2 Author identity

Author identity is also given in SCALE 1.0. We thus define AUTHORID, four binary features representing each of the four authors in the SCALE 1.0 corpus. We use a 'one-hot' representation: the single feature corresponding to the particular author is active when it is that author who wrote the review in which the gap occurs, and the other three features inactive.

6.3 Results and discussion

Results comparing various combinations of the above features with unigram features are shown in Table 7 as an accuracy rating and a percentage increase over the unigram baseline. Features were only tested in conjunction with DOCPRES and SENTPRES, as the better performing unigram baselines in Section 5.

Table 7 shows that for the most part these features have little to no impact on prediction accuracy. However, AUTHORID (whether by itself or in conjunction with DOCSENT) always produces the best results. In three of the cases it is only by a small margin, with the exception being the case of AUTHORID on the SENTENCE FREQUENCY classifier. The addition of AUTHORID to the SENTENCE PRESENCE unigrams, which outperform DOCUMENT PRESENCE unigrams on the other word set classes, comes closest to approximating the DOCUMENT PRESENCE results on the same affect word sets.

To look further into our intuition that knowledge of the author reflects the linguistic creativity discussed above, we examined the impact of a DOCUMENT FREQUENCY (DF) threshold (Yang & Pedersen, 1997). A major conclusion of Wiebe et al. (2004) was that rare events such as *hapax legomena* contain a lot of information for subjective texts, and so feature selection such as by DF thresholding would be harmful. Therefore, we might expect that DF thresholding would worsen results here if this linguistic creativity is what has led to the SAME-AFFECT results. For this, we examined the development set (which has similar overall results on the various classifiers). Using DOCUMENT FREQUENCY thresholding values of 2, 4 and 8 caused small decreases (typically around +0.1% at a DF of 8) in the performance of all of the features above. The small magnitude of these suggested that the source of the SAME-AFFECT results might be elsewhere.

If we consider a practical task, rather than the situation above where we are just investigating the nature of the influence of the author, we would likely not know the author, and have to approximate author identity in order for the method to be effective on unseen text by unknown authors. Given the small magnitude of the improvement, this would likely be overwhelmed by errors in predicting the author, and so not a practical suggestion for such a task. We therefore turn our attention to an alternative way of looking at broader document context.

Additional features	Overall dev.	Overall test	Overall both	No affect dev.	No affect test	No affect both	Same affect dev.	Same affect test	Same affect both	Differing affect dev.	Differing affect test	Differing affect both
None (unigram only)												
DocPres	77·6	75·8	75·6	82·8	87·3	84·9	58·4	**73·8**	**72·8**	67·3	58·6	62·6
SentPres	81·7	78·6	80·0	85·5	89·6	87·4	56·5	73·1	72·0	**75·1**	65·9	70·1
AuthorID												
DocPres	+0·0	+0·0	+0·0	–	+0·0	+0·0	+0·3	−0·0	0·0	+0·0	+0·0	+0·0
SentPres	**+0·0**	+0·4‡	**+0·3‡**	**+0·1**	**+0·1**	**+0·1**	**+1.3**	+0·8‡	+0·8‡	−0·2	+0·6⋆	**+0·3**
MicroWNOP												
DocPres	+0·0	+0·0	+0·0	−0·1	−0·1	−0·1⋆	−1·0⋆	−0·1	−0·1	+0·4⋆	+0·4⋆	+0·4†
SentPres	−1·4‡	−1·5‡	−1·4‡	−0·3†	−1·1‡	−0·7‡	−1·9	−0·1	−0·2	−3·8‡	−3·4‡	−3·6‡
DocSent												
DocPres	+0·0	−0·0	+0·0	–	–	–	–	+0·0	+0·0	+0·1	+0·0	+0·0
SentPres	−0·1⋆	+0·1	−0·0	−0·0	−0·0	−0·0	+0·3	+0·0	+0·0	−0·4⋆	+0·3⋆	−0·0
MicroWNOP and AuthorID												
DocPres	+0·0	+0·0	+0·0	−0·1	−0·1	−0·1	−1·0⋆	−0·0	−0·1	+0·4⋆	+0·3	+0·3⋆
SentPres	−1·4‡	−0·8‡	−1·1‡	−0·3†	−0·9‡	−0·6‡	+0·6	+0·8‡	+0·8‡	−3·9‡	−2·4‡	−3·1‡
DocSent and AuthorID												
DocPres	+0·0	+0·0	+0·0	–	+0·0	+0·0	**+0·0**	−0·0	−0·0	+0·1	+0·1	+0·1
SentPres	−0·1	**+0·5‡**	+0·3‡	+0·0	+0·1	+0·1	+1·9	+0·9‡	+1·0‡	−0·5†	**+0·8†**	+0·2

Bold values are best performance for that column

A − means model's choice of word was identical to the unigram PRESENCE model, +0.0 means an increase of < 0.05, −0.0 means a decrease with absolute value < 0.05

⋆ Difference from unigram model significant at the $p < 0.05$ level

† Difference from unigram model significant at the $p < 0.01$ level

‡ Difference from unigram model significant at the $p < 0.001$ level

Table 7. *Performance of SVMs using unigrams with additional features, compared to unigram* PRESENCE *models*

7 Unigram models accounting for distance

7.1 Distance measure

Our finding in Section 5 that SENTENCE PRESENCE features usually outperform DOCUMENT PRESENCE features coheres with the finding of Edmonds (1997) that a very small window around the data provided better choice accuracy. (Subsequent authors have tended not to evaluate very wide windows in the first place.) Presumably, the noise in the more distant words overwhelms any useful information they convey. However, the finding for the SAME-AFFECT set that DOCUMENT PRESENCE improves performance hints that document-level features can have an impact on the FITB task at least in some cases. This has parallels in work on the behaviour of context with respect to entropy, whose underlying principle as described by Qian and Jaeger (2010) is "that distant contextual cues tend to gradually lose their relevance for predicting upcoming linguistic signals". This idea was first presented in the context of an exploration of the behaviour of entropy by Genzel and Charniak (2002), where they propose the Constant Entropy Rate principle. By conditionally decomposing entropy with respect to local (i.e. sentence-level) and broader context, they empirically demonstrate support — through the consistently observed increase in entropy conditioned on local context throughout texts — for their principle, and conclude that broader context continues to influence later text. They develop this further in Genzel and Charniak (2003), where they find changes in entropy behaviours at paragraph boundaries, suggesting topic or other broader context characteristics are part of the effect. Their principle has subsequently been supported by work in psycholinguistics, such as that of Keller (2004), Levy and Jaeger (2007), and Gallo, Jaeger, and Smyth (2008). Qian and Jaeger (2010) go on to investigate the precise type of relationship of broader context, and find that incorporating linear and sub-linear representations of broader context into entropy models improves the fit of these models of the development of entropy throughout a text.

As noted in Section 5, work in the structurally similar task of WSD has generally ignored broader context. Prompted by the results in the previous section and the work on discourse entropy, in this section we describe another model, in which unigram features are weighted by their distance from the gap. Weighting feature values has been explored in a number of machine learning contexts, often drawing on Information Retrieval tools — Paltoglou and Thelwall (2010) is an example, investigating various weighting schemes in the context of document sentiment analysis — but whereas these are typically for whole-document classification and weight feature values identically within a document (e.g. by tf.idf), our weighting scheme is a function of distance.

Rather than using feature value 1 for PRESENCE or 0 for ABSENCE, as in the unigram models in Section 5, here we weight the presence of a token by its distance from the gap. For example, in the sentence fragment in (10) the distance of the token *big* from the gap is 1, and the distance of *economists* from the gap is 7.

(10) ... some economists think it would be a big _____.

In order that features further away from the gap not be entirely eliminated by their distance from the gap, but that noise not overwhelm the information they bring, we experiment with weighting the distance using the following functions for the feature value $f(w)$ of a token w using distance $d(g, w)$ in number of tokens between w and gap g:

- Inverse linear weighting, InvLinear:

$$f(w) = \frac{1}{d(g, w)} \tag{11}$$

- Inverse square root weighting, InvSquareRoot:

$$f(w) = \frac{1}{\sqrt{d(g, w)}} \tag{12}$$

As in Section 5, every token in the document is considered as a feature, except those with total corpus frequency of less than 4 and the candidates to fill the gap themselves. If a token w is used more than once in a document, the largest value for $f(w)$, ie the smallest $d(g, w)$ for both InvLinear and InvSquareRoot for a given gap g is used.

We test the effectiveness of limiting the features to text surrounding the target word, using the following measures:

- every token in the document;
- every token in the sentence containing the gap and the two surrounding sentences; and
- every token in the sentence containing the gap, only.

We also test the InvSquareRoot unigram model with selected successful additional features from Section 6, giving us the following models:

1. InvSquareRoot
2. InvSquareRoot and AuthorID combined
3. InvSquareRoot and DocSent combined
4. InvSquareRoot, AuthorID and DocSent combined.

7.2 Results and discussion

Results are shown in Table 8. We see that weighting the unigrams for their distance from the gap is useful in all cases, giving very large relative improvements over the baseline (in error reduction, up to 48% for differing-affect sets with InvLinear). This general pattern accords with the behaviour of broader context in the entropy work discussed earlier.

While this result is to some extent expected — words closer to the gap have a higher weight, and thus the most predictive power over the word filling the gap — the most interesting result is that a wider context than sentence level remains useful. The result even extends beyond the 3-sentence level to the entire document, so that in at least some cases words quite far from the gap indeed are affecting the choice of near synonym: drilling down into the data for examples, the improvement

Distance measure and span	Overall dev.	Overall test	Overall both	No affect dev.	No affect test	No affect both	Same affect dev.	Same affect test	Same affect both	Differing affect dev.	Differing affect test	Differing affect both
None												
Document	77·6	75·8	76·6	82·8	87·3	84·9	58·4	73·8	72·8	67·3	58·6	62·6
3 sentence	80·5	76·7	78·4	85·1	88·8	86·8	56·5	72·2	71·1	71·8	61·2	66·1
1 sentence	81·7	78·6	80·0	85·5	89·6	87·4	56·5	73·1	72·0	75·1	65·9	62·6
InvLinear												
Document	**+10·1‡**	**+7·8‡**	**+8·8‡**	**+7·1‡**	**+5·3‡**	**+6·3‡**	+2·6	+0·9⋆	+1·1⋆	**+17·7‡**	**+19·0‡**	**+18·4‡**
3 sentence	+7·1‡	+6·8‡	+6·9‡	+4·7‡	3·8‡	+4·3‡	+4·5⋆	**+2·3‡**	**+2·4‡**	+12·7‡	+16·4‡	+14·7‡
1 sentence	+5·9‡	+5·0‡	+5·4‡	+4·4‡	+3·0‡	+3·8‡	+4·2⋆	+1·7‡	+1·9‡	+9·6‡	+11·9‡	+10·8‡
InvSquareRoot												
Document	+7·6‡	+6·5‡	+7·0‡	+4·8‡	+4·3‡	+4·6‡	+3·5⋆	+1·2†	+1·4‡	+14·2‡	+15·7‡	+15·0‡
3 sentence	+5·2‡	+5·5‡	+5·4‡	+3·1‡	+2·9‡	+3·0‡	**+7·1‡**	+1·8‡	+2·2‡	+9·9‡	+13·5‡	+11·8‡
1 sentence	+4·6‡	+4·2‡	+4·4‡	+3·3‡	+2·5‡	+3·0‡	+6·8‡	+1·9‡	+2·2‡	+7·3‡	+9·5‡	+8·5

Bold values are best performance for that column

⋆ Difference from non-distance weighted unigram model significant at the $p < 0.05$ level

† Difference from non-distance weighted unigram model significant at the $p < 0.01$ level

‡ Difference from non-distance weighted unigram model significant at the $p < 0.001$ level

Table 8. *Performance of SVMs using distance measures, compared to unigram* PRESENCE *models*

for the set *ludicrous, senseless, foolish, ...* with document-level context is +11.2% over the baseline versus +6.6% for three-sentence context (both InvSquareRoot); the improvement for *aloof, detached, reserved* is +17.2% for document-level context versus +13.1% for three-sentence context (again InvSquareRoot). As with the unigram results from Section 5, this supports our hypothesis that features of the entire document influence the choice of near synonym, even though document-level features do not appear to have been well captured by our choice of features in Section 6.

One possible reason for this is connected to empirical work by, for example, Bieler, Dipper, and Stede (2007), which has found that movie (and other similar) reviews have a semi-conventionalised structure: there are distinguishable "functional zones", with more objective descriptive material tending to cluster together separately from more subjective commentary. Perhaps the weighting is implicitly giving more prominence to the context in the same functional zone.[17]

Same-affect near synonyms also respond better to a different weighting function, InvSquareRoot rather than InvLinear. InvSquareRoot discounts the distance between a word and the gap less heavily than InvLinear does, especially, relatively speaking, at more extreme distances. This further demonstrates that more distant words are having an effect: some discounting is evidently required since InvSquareRoot outperforms no weighting, but there is an extent past which the discounting appears to under-weight features when the entire document is required as context.

As for the general utility of the weighting functions, having no weighting function results in the document context performing 3.4% worse than the single sentence context: this is consistent with the discussion in Section 5.2 on previous and related results, where the document context just adds noise. Weighting functions boost the results for both the single sentence context and the document context; for the no-affect and differing-affect sets and the InvLinear function, this is to the same (highest performing) level. The weighting function thus seems like a good way of ignoring noise, although it does not entirely compensate for extending the context beyond what is necessary.

Results for the combination of the InvLinear weighting with other features is given in Table 9. In all of these cases, adding the features was harmful. Distance weighting thus appears to capture document tone better than the explicit features of author ID or document sentiment.

We have thus further confirmed our result that in many cases, a very wide context is useful, and we have gone some way to narrowing down exactly how to balance providing this context with weighting appropriately for noise. Further work is needed to determine how to distinguish contexts where words very distant from the gap should be included with appropriate weights, as in our same-affect set, and where they should be excluded entirely, or weighted even lower than InvLinear.

[17] We thank an anonymous reviewer for this insight.

Additional features	Overall	No affect	Same affect	Differing affect
None	76·6	84·9	72·8	62·6
InvLinear	**+8·8**	**+6·3**	**+1·1**	**+18·4**
InvLinear and AuthorID	−1·3	−0·4	−1·2	−3·1
InvLinear and DocSent	−1·3	−0·4	−1·2	−3·1
InvLinear, AuthorID and DocSent	−1·3	−0·4	−1·2	−3·1

Bold values are the best performance for that column.

Table 9. *Performance of SVMs using* InvLinear *with additional features, compared to the* DocPres *unigram model*

7.3 Examples of the best performing sets

As an illustration, the best performing five near synonym sets relative to improvement over the baseline — that is, those where broader context and distance-weighting were particularly useful, which wouldn't be captured by more typical small context window sizes — are shown in Table 10 for both document InvLinear and sentence InvLinear. For both techniques, the largest improvements are in fact delivered over the full range of set types, including no affect. The only difference is the appearance of different same-affect sets in the top five: *aghast* etc in the document list and *brashness* etc in the sentence list, with each being seventh in the other list.

8 Conclusion

In this paper, we have applied a previously untried supervised approach to the problem of choosing the right near synonym to fill a lexical gap. Our main conclusions from doing this are as follows. First, as per Pang et al. (2002) with document sentiment classification, unigrams alone do well, with presence outperforming frequency, and with immediate-context (sentence) models generally outperforming wider-context (document) models. Second, that once appropriate weighting of distance features are incorporated, the technique performs near-synonym choice notably better with document rather than sentence features; this may be a consequence of a particular 'tone' suffusing the document. Third, adding in one possible factor related to this tone, knowledge of the author of the text, gives slightly better results overall, in particular improving sentence presence results for same-affect near synonyms; this author effect was not expected at the start of the work. Fourth, the most significant improvement came from incorporating document-level information using a weighting scheme, which in fact improved over the earlier best sentence-level models, and which in its description of the effect of broader context mirrors work on the effect of the Constant Entropy Rate principle. This is true for all near-synonym sets, including the non-affect ones.

Words in set	Affect type	No. tests	Performance	Improvement
		Document InvLinear		
recommendation, *advice*	NO-AFFECT	146	91·8%	+37·7
feat, *operation*, *act*, *exploit*, *action*, *performance*	DIFFERING-AFFECT	3596	85·6%	+35·5
charge, *attack*, *storm*, *assault*	NO-AFFECT	177	65·0%	+33·9
precise, *accurate*, *exact*, *right*, *nice*, *correct*, *true*	DIFFERING-AFFECT	2181	76·9%	+30·7
aghast, *scared*, *frightened*, *afraid*	SAME-AFFECT	187	67·4%	+26·2
		Sentence InvLinear		
recommendation, *advice*	NO-AFFECT	146	90·4%	+36·3
feat, *operation*, *act*, *exploit*, *action*, *performance*	DIFFERING-AFFECT	3596	85·2%	+35·1
charge, *attack*, *storm*, *assault*	NO-AFFECT	177	62·7%	+31·6
precise, *accurate*, *exact*, *right*, *nice*, *correct*, *true*	DIFFERING-AFFECT	2181	77·1%	+30·9
brashness, *brass*, *cheek*, *hide*, *nerve*	SAME-AFFECT	70	60·0%	+25·7

Table 10. *Best performing five sets for each of document and sentence* InvLinear*, relative to baseline performance*

There are a number of directions for future work. In terms of the specific task tackled in this article, there could be more investigation into how precisely this author and distance information are causing the improvement discovered, and whether more sophisticated models could capture this. In terms of alternative approaches, dependency relations could be promising, either in terms of structured language models (Xu, Chelba, & Jelinek, 2002) or as additional features in the supervised model (Özgür & Güngör, 2010). In terms of applications, the techniques in the article could be applied to the sentiment-directed text rewriting noted at the start of the article described in (Inkpen et al., 2006), or the intelligent thesaurus of (Inkpen, 2007a). In both of these applications, as well as the FITB task of this article, it would be useful to devise and carry out human evaluations — for FITB, for ex-

ample, alternatives besides the original choice might be acceptable — which could feasibly be done with the availability of services like Mechanical Turk.

Acknowledgements

The authors would like to thank the anonymous reviewers of the article, and to acknowledge the support of ARC Discovery grant DP0558852.

A Word sets with sentiment differences

These are the 47 word sets drawn from *Use the Right Word* (Hayakawa, 1968) as described in Section 3.2. The annotation Positive, Neutral or Negative is given for each word; sets are annotated as None (no affect), Same (same affect) or Differing (differing affect). The number of instances of each set is given under Size.

Set sentiment type	Size	Word	Sentiment	Word	Sentiment	Word	Sentiment
None	34	*incorporate*	Neutral	*digest*	Neutral	*absorb*	Neutral
Same	988	*ludicrous*	Negative	*senseless*	Negative	*foolish*	Negative
		preposterous	Negative	*ridiculous*	Negative	*farcical*	Negative
		absurd	Negative	*silly*	Negative	*irrational*	Negative
		unreasonable	Negative				
None	21	*attend*	Neutral	*accompany*	Neutral		
None	23	*collect*	Neutral	*gather*	Neutral		
Differing	2181	*precise*	Neutral	*accurate*	Neutral	*exact*	Neutral
		right	Positive	*nice*	Neutral	*correct*	Neutral
		true	Neutral				
Differing	163	*acknowledge*	Neutral	*confess*	Negative	*admit*	Neutral
Differing	3596	*feat*	Positive	*operation*	Neutral	*act*	Neutral
		exploit	Positive	*action*	Neutral	*performance*	Neutral
None	44	*activity*	Neutral	*stir*	Neutral		
Differing	129	*insight*	Positive	*perception*	Neutral		
None	149	*fit*	Neutral	*conform*	Neutral	*adapt*	Neutral
None	132	*supplement*	Neutral	*addition*	Neutral		
None	2113	*adequate*	Neutral	*satisfactory*	Neutral	*enough*	Neutral
		sufficient	Neutral				
None	146	*recommendation*	Neutral	*advice*	Neutral		
Same	187	*aghast*	Negative	*scared*	Negative	*frightened*	Negative
		afraid	Negative				
Differing	32	*drunk*	Negative	*alcoholic*	Neutral		
Same	33	*fidelity*	Positive	*loyalty*	Positive		
None	77	*fable*	Neutral	*allegory*	Neutral		
Differing	99	*aloof*	Negative	*reserved*	Positive	*detached*	Negative
None	690	*old*	Neutral	*ancient*	Neutral		
Same	123	*indignation*	Negative	*rage*	Negative	*wrath*	Negative
		fury	Negative	*anger*	Negative		
Differing	198	*creature*	Neutral	*animal*	Neutral	*beast*	Negative
None	724	*reply*	Neutral	*response*	Neutral	*answer*	Neutral
Same	177	*foreboding*	Negative	*anxiety*	Negative	*angst*	Negative
		worry	Negative	*dread*	Negative		
None	2357	*aspect*	Neutral	*look*	Neutral	*appearance*	Neutral
None	28	*acclaim*	Neutral	*applause*	Neutral		
None	7123	*around*	Neutral	*about*	Neutral	*approximately*	Neutral
		roughly	Neutral				
Differing	806	*debate*	Neutral	*discuss*	Positive	*reason*	Neutral
		argue	Neutral				

None	818	*result*	Neutral	*issue*	Neutral	*stem*	Neutral
		emerge	Neutral	*arise*	Neutral		
None	716	*material*	Neutral	*weapons*	Neutral	*arms*	Neutral
Differing	158	*synthetic*	Neutral	*ersatz*	Negative	*false*	Negative
		imitation	Neutral				
Differing	152	*stylist*	Positive	*artist*	Neutral	*creator*	Neutral
		virtuoso	Positive	*painter*	Neutral		
Differing	344	*mannered*	Negative	*artificial*	Negative	*artistic*	Neutral
		precious	Negative	*arty*	Negative	*stylized*	Positive
		aesthetic	Positive				
None	104	*ally*	Neutral	*associate*	Neutral	*fellow*	Neutral
		partner	Neutral				
None	209	*promise*	Neutral	*guarantee*	Neutral	*assure*	Neutral
None	177	*charge*	Neutral	*attack*	Neutral	*storm*	Neutral
		assault	Neutral				
None	205	*sympathy*	Neutral	*attraction*	Neutral	*affinity*	Neutral
Differing	275	*credit*	Positive	*attribute*	Neutral		
Same	1750	*bad*	Negative	*distasteful*	Negative	*objectionable*	Negative
		unpleasant	Negative				
Same	163	*banal*	Negative	*fatuous*	Negative	*inane*	Negative
		inspid	Negative	*vapid*	Negative		
Same	147	*bait*	Negative	*hector*	Negative	*hound*	Negative
		ride	Negative				
Same	54	*bigotry*	Negative	*bias*	Negative	*intolerance*	Negative
		prejudice	Negative				
Same	23	*bitterness*	Negative	*harshness*	Negative		
Same	118	*bleak*	Negative	*barren*	Negative	*desolate*	Negative
		gaunt	Negative				
Same	70	*brashness*	Negative	*brass*	Negative	*cheek*	Negative
		hide	Negative	*nerve*	Negative		
Same	546	*abrupt*	Negative	*curt*	Negative	*gruff*	Negative
		short	Negative				
Same	146	*catastrophe*	Negative	*debacle*	Negative	*disaster*	Negative
Same	224	*clumsy*	Negative	*awkward*	Negative	*gawky*	Negative
		inept	Negative	*lumbering*	Negative		

B Language models

Here we detail the implementation of our language model prediction system using the Web1T data in order to test whether our methods out-perform language models. We consider predicting a word choice as estimating the most probable 3-gram. In example 1 we would be trying to estimate the most probable of the following 3-grams:

(13) a big *error*

(14) a big *mistake*

(15) a big *oversight*

We would then choose from among *error*, *mistake* and *oversight* by choosing the word contained in the most probable 3-gram.

As is common in language models, we back off to 2-grams and 1-grams where necessary. We draw our discussion that follows from the original paper of Katz (1987), the more detailed explication of Gale and Sampson (1995), and the overview of Jurafsky and Martin (2009). In general, backoff models look like this:

- The count of ngram $w_1, \ldots, w_n$ in the training data is denoted by

$$C(w_1, \ldots, w_n) \tag{16}$$

- The smoothed count C^* of ngram $w_1, \ldots, w_n$ is given by the specific smoothing algorithm.
- The adjusted probability P^* of ngram $w_1, \ldots, w_n$ is given by

$$P^*(w_n|w_1, \ldots, w_{n-1}) = \frac{C^*(w_1, \ldots, w_n)}{C(w_1, \ldots, w_{n-1})} \tag{17}$$

- If a count for $w_1, \ldots, w_n$ is unavailable, $P(w_n|w_1, \ldots, w_{n-1})$ is estimated using the counts for ngram $w_2, \ldots, w_n$ using a back-off model, where α is a the proportion of the probability space reserved for unseen events:

$$P(w_n|w_1, \ldots, w_{n-1}) = \alpha(w_1, \ldots, w_{n-1})P(w_n|w_2, \ldots, w_{n-1}) \tag{18}$$

In the smoothing and backoff implementation of Katz (1987) the specifics of these functions are:

- The number of ngrams with count c in the training data is denoted by

$$r(c) \tag{19}$$

- The adjusted count C^* of an ngram $w_1, \ldots, w_n$ is smoothed by the Good-Turing estimation (Good, 1953) and is given by

$$C^*(w_1, \ldots, w_n) = \frac{(C(w_1, \ldots, w_n) + 1) \cdot r(C(w_1, \ldots, w_n) + 1)}{r(C(w_1, \ldots, w_n))} \tag{20}$$

- The proportion α of the total probability mass allocated to unseen words w_n following $w_1, \ldots, w_{n-1}$ is given by:

$$\alpha(w_1, \ldots, w_{n-1}) = \frac{\beta(w_1, \ldots, w_{n-1})}{\sum_{w_n : C(w_2, \ldots, w_n) > 0} P^*(w_n|w_2, \ldots, w_{n-1})} \tag{21}$$

where the function β is given by:

$$\beta(w_1, \ldots, w_{n-1}) = 1 - \sum_{w_n : C(w_1, \ldots, w_n) > 0} P^*(w_n|w_1, \ldots, w_{n-1}) \tag{22}$$

In a typical language model implementation, at some sufficiently large value of r the probability of n-gram $w_1, \ldots, w_n$, $P(w_n|w_1, \ldots, w_{n-1})$ would be estimated using a maximum likelihood estimate instead, given in equation 23, as in for example Gale and Sampson (1995):

$$P(w_n|w_1, \ldots, w_{n-1}) = \frac{C(w_1, \ldots, w_n)}{C(w_1, \ldots, w_{n-1})} \tag{23}$$

Given that we are using Web1T for our values of $C(w_1, \ldots, w_n)$, and Web1T does not provide counts for $r < 40$ for bi- to 5-grams and for $r < 200$ for we use equation 23 rather than equation 17. There are several reasons for this. The first is that the standard methods for determining the value of this cut-off would usually be applied on the tail end of the data, precisely what's been removed from Web1T.

The second is that even for fairly low values of r available in Web1T, ie $r \approx 40$, $C^*(w_1, \ldots, w_n) > C(w_1, \ldots, w_n)$ when using equation 20 to compute C^*. This results in periodic cases where $\beta < 0$ in equation 22.

Given this, we also estimate α differently from equation 21. The Web1T corpus does not include n-grams for $n \geq 2$ with counts of less than 40 (or unigrams with counts of less than 200). We therefore estimate the probability mass allocated to unseen n-grams by the proportion of the count of an n-gram $w_1, \ldots, w_n$ unaccounted for by known $n+1$-grams $w_1, \ldots, w_{n+1}$. To give an example of how this is done, let us assume that bigram *mistakes are* has a Web1T count of 300, and the only trigrams beginning with *mistakes are* are *mistakes are bad* and *mistakes are good*, with counts of 75 each. We then have 150 unseen tokens following *mistakes are* and thus the value of α is 0.5.

This can be formally expressed as

$$\alpha(w_1, \ldots, w_{n-1}) = \frac{\sum_{w_n : C(w_1, \ldots, w_n) > 0} C(w_1, \ldots, w_n)}{C(w_1, \ldots, w_{n-1})} \tag{24}$$

References

Banerjee, S., & Pedersen, T. (2003). The Design, Implementation and Use of the Ngram Statistics Package. In *Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics* Mexico City.

Bieler, H., Dipper, S., & Stede, M. (2007). Identifying formal and functional zones in film reviews. In *Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue*, pp. 75–85.

Brants, T., & Franz, A. (2006). Web 1T 5-gram Version 1.. `http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2006T13`.

Carbonell, J., Klein, S., Miller, D., Steinbaum, M., Grassiany, T., & Frei, J. (2006). Context-Based Machine Translation. In *Proceedings of the 7th Conference of the Association for Machine Translation of the Americas (AMTA)*, pp. 19–28.

Cerini, S., Compagnoni, V., Demontis, A., Formentelli, M., & Gandi, C. (2007). Micro-WNOp: A Gold Standard for the evaluation of automatically compiled lexical resources for Opinion Mining. In *Language resources and linguistic theory: Typology, second language acquisition, English linguistics.* Franco Angeli, Milan, Italy.

Church, K., Gale, W., Hanks, P., & Hindle, D. (1989). Parsing, Word Associations and Typical Predicate-Argument Relations. In *Proceedings of the International Workshop on Parsing Technologies.*

Church, K., & Hanks, P. (1991). Word Association Norms, Mutual Information and Lexicography. *Computational Linguistics*, *16*(1), 22–29.

Church, K., Gale, W., Hanks, P., & Hindle, D. (1991). Using statistics in lexical analysis. In Zernick, U. (Ed.), *Lexical Acquisition: Using On-line Resources to Build a Lexicon*, pp. 115–164. Lawrence Erlbaum Associates.

Clarke, C. L. A., & Terra, E. L. (2003). Passage retrieval vs. document retrieval

for factoid question answering. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 427–428 Toronto, Canada.

DiMarco, C., Hirst, G., & Stede, M. (1993). The semantic and stylistic differentiation of synonyms and near-synonyms. In *Proceedings of AAAI Spring Symposium on Building Lexicons for Machine Translation*, pp. 114–121 Stanford, CA, USA.

Edmonds, P. (1997). Choosing the Word Most Typical in Context Using a Lexical Co-occurrence Network. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, pp. 507–509 Madrid, Spain. Association for Computational Linguistics.

Edmonds, P. (1999). *Semantic representations of near-synonyms for automatic lexical choice*. Ph.D. thesis, University of Toronto.

Edmonds, P., & Hirst, G. (2002). Near-synonymy and lexical choice. *Computational Linguistics*, *28*(2), 105–144.

Esuli, A., & Sebastiani, F. (2006). SentiWordNet: A Publicly Available Lexical Resource for Opinion Mining. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC 2006)*, pp. 417–422 Genova, Italy.

Fellbaum, C. (Ed.). (1998). *WordNet: An Electronic Lexical Database*. The MIT Press.

Foody, G. M. (2008). Sample Size Determination for Image Classification Accuracy Assessment and Comparison. In *Proceedings of the 8th International Symposium on Spatial Accuracy Assessment in Natural Resources and Environmental Sciences*, pp. 154–162 Shanghai.

Gale, W. A., & Sampson, G. (1995). Good-Turing Frequency Estimation Without Tears. *Journal of Quantitative Linguistics*, *2*, 217–232.

Gallo, C. G., Jaeger, T. F., & Smyth, R. (2008). Incremental Syntactic Planning across Clauses. In *Proceedings of the 30th Annual Meeting of the Cognitive Science Society (CogSci08)*, pp. 845–850.

Gamon, M. (2004). Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis. In *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*, pp. 841–847 Morristown, NJ, USA. Association for Computational Linguistics.

Gardiner, M., & Dras, M. (2007a). Corpus Statistics Approaches to Discriminating Among Near-Synonyms. In *Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics (PACLING 2007)*, pp. 31–39 Melbourne, Australia.

Gardiner, M., & Dras, M. (2007b). Exploring Approaches to Discriminating among Near-Synonyms. In *Proceedings of the Australasian Language Technology Workshop 2007*, pp. 31–39 Melbourne, Australia.

Genzel, D., & Charniak, E. (2002). Entropy Rate Constancy in Text. In *Proceedings of the 40th Annual Meetings of the Association for Computational Linguistics (ACL'02)*, pp. 199–206 Philadelphia, US.

Genzel, D., & Charniak, E. (2003). Variation of Entropy and Parse Trees of Sentences as a Function of the Sentence Number. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 65–72 Sapporo, Japan.

Good, I. J. (1953). The population frequencies of species and the estimation of population parameters. *Biometrika*, *40*(3–4), 237–264.

Hassan, S., Csomai, A., Banea, C., Sinha, R., & Mihalcea, R. (2007). UNT: SubFinder: Combining Knowledge Sources for Automatic Lexical Substitution. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pp. 410–413 Prague, Czech Republic. Association for Computational Linguistics.

Hatzivassiloglou, V., & Wiebe, J. M. (2000). Effects of Adjective Orientation and Gradability on Sentence Subjectivity. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING-2000)*, pp. 299–305 Saarbrcken, Germany.

Hawker, T., Gardiner, M., & Bennetts, A. (2007). Practical Queries of a Massive n-gram Database. In *Proceedings of the Australasian Language Technology Workshop 2007*, pp. 40–48 Melbourne, Australia.

Hayakawa, S. I. (Ed.). (1968). *Use The Right Word: Modern Guide to Synonyms and Related Words* (1st edition). The Reader's Digest Association Pty. Ltd.

Hayakawa, S. I. (Ed.). (1994). *Choose the Right Word* (2nd edition). Harper Collins Publishers. revised by Eugene Ehrlich.

Ide, N., & Vronis, J. (1998). Introduction to the Special Issue on Word Sense Disambiguation: The State of the Art. *Computational Linguistics*, *24*(1), 1–40.

Inkpen, D. (2007a). Near-Synonym Choice in an Intelligent Thesaurus. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pp. 356–363 Rochester, New York. Association for Computational Linguistics.

Inkpen, D. (2007b). A statistical model for near-synonym choice. *ACM Transactions of Speech and Language Processing*, *4*(1), 1–17.

Inkpen, D., & Hirst, G. (2006). Building and using a lexical knowledge-base of near-synonym differences. *Computational Linguistics*, *32*(2), 223–262.

Inkpen, D. Z., Feiguina, O., & Hirst, G. (2006). Generating more-positive or more-negative text. In Shanahan, J. G., Qu, Y., & Wiebe, J. (Eds.), *Computing Attitude and Affect in Text (Selected papers from the Proceedings of the Workshop on Attitude and Affect in Text, AAAI 2004 Spring Symposium)*, pp. 187–196. Springer, Dordrecht, The Netherlands.

Islam, A., & Inkpen, D. (2010). Near-Synonym Choice using a 5-gram Language Model. *Research in Computing Science: Special issue on Natural Language Processing and its Applications*, *46*, 41–52.

Islam, M. A. (2011). *An Unsupervised Approach to Detecting and Correcting Errors in Text*. Ph.D. thesis, University of Ottawa.

Joachims, T. (1999). Making large-Scale SVM Learning Practical. In Schlkopf, B.,

Burges, C. J., & Smola, A. J. (Eds.), *Advances in Kernel Methods - Support Vector Learning*, pp. 169–184. The MIT Press, Cambridge, USA.

Jurafsky, D., & Martin, J. H. (2009). *Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition, and Computational Linguistics* (2nd edition). Prentice-Hall.

Katz, S. M. (1987). Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics, Speech and Signal Processing*, *35*, 400–401.

Keller, F. (2004). The Entropy Rate Principle as a Predictor of Processing Effort: An Evaluation against Eye-tracking Data. In Lin, D., & Wu, D. (Eds.), *Proceedings of EMNLP 2004*, pp. 317–324 Barcelona, Spain. Association for Computational Linguistics.

Koppel, M., Akiva, N., & Dagan, I. (2006a). Feature Instability as a Criterion for Selecting Potential Style Markers. *Journal of the American Society for Information Science and Technology*, *57*(11), 1519–1525.

Koppel, M., Akiva, N., & Dagan, I. (2006b). Feature Instability as a Criterion for Selecting Potential Style Markers. *Journal of the American Society for Information Science and Technology*, *57*(11), 1519–1525.

Kullback, S., & Leibler, R. A. (1951). On Information and Sufficiency. *The Annals of Mathematical Statistics*, *22*, 79–86.

Landauer, T., & Dumais, S. (1997). A solution to Plato's problem: the latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, *104*(2), 211–240.

Langkilde, I., & Knight, K. (1998). The practical value of N-grams in generation. In *Proceedings of the 9th International Natural Language Generation Workshop*, pp. 248–255 Niagra-on-the-Lake, Canada.

Levy, R., & Jaeger, T. F. (2007). Speakers optimize information density through syntactic reduction. In Schlkopf, B., Platt, J., & Hoffman, T. (Eds.), *Advances in Neural Information Processing Systems 19*. MIT Press, Cambridge, MA.

Liu, Y., & Zheng, Y. F. (2005). One-against-all multi-class SVM classification using reliability measures. In *Proceedings of the 2005 IEEE International Joint Conference on Neural Networks, 2005. (IJCNN '05.)*, Vol. 2, pp. 849–854.

McCarthy, D., & Navigli, R. (2007). SemEval-2007 Task 10: English Lexical Substitution Task. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pp. 48–53 Prague, Czech Republic. Association for Computational Linguistics.

Özgür, L., & Güngör, T. (2010). Text classification with the support of pruned dependency patterns. *Pattern Recognition Letters*, *31*(12), 1598–1607.

Paltoglou, G., & Thelwall, M. (2010). A Study of Information Retrieval Weighting Schemes for Sentiment Analysis. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pp. 1386–1395 Uppsala, Sweden. Association for Computational Linguistics.

Pang, B., & Lee, L. (2004). A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pp. 271–278 Barcelona, Spain.

Pang, B., & Lee, L. (2005). Seeing Stars: Exploiting Class Relationships for Sentiment Categorization with Respect to Rating Scales. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pp. 115–124 Ann Arbor, Michigan. Association for Computational Linguistics.

Pang, B., & Lee, L. (2008). Opinion Mining and Sentiment Analysis. *Foundations and Trends in Information Retrieval*, *2*(1–2), 1–135.

Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up? Sentiment Classification using Machine Learning Techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, pp. 79–86. Association for Computational Linguistics.

Qian, T., & Jaeger, T. F. (2010). Close = Relevant? The Role of Context in Efficient Language Production. In *Proceedings of the 2010 Workshop on Cognitive Modeling and Computational Linguistics*, pp. 45–53 Uppsala, Sweden. Association for Computational Linguistics.

Rapp, R. (2008). The automatic generation of thesauri of related words for English, French, German, and Russian. *International Journal of Speech Technology*, *11*(3–4), 147–156.

Refaeilzadeh, P., Tang, L., & Liu, H. (2009). Cross Validation. In Tamer, M., & Liu, L. (Eds.), *Encyclopedia of Database Systems*. Springer.

Reiter, E., & Dale, R. (2000). *Building Natural Language Generation Systems*. Cambridge University Press.

Rifkin, R., & Klautau, A. (2004). In Defense of One-Vs-All Classification. *Journal of Machine Learning Research*, *5*(2), 101–141.

Rosen-Zvi, M., Griffiths, T., Steyvers, M., & Smyth, P. (2004). The author-topic model for authors and documents. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, pp. 487–494.

Salzberg, S. L. (1997). On Comparing Classifiers: Pitfalls to Avoid and a Recommended Approach. *Data Mining and Knowledge Discovery*, *1*, 317–327.

Sinclair, J. (1987). The Nature of the Evidence. In Sinclair, J. M. (Ed.), *Looking Up: An Account of the COBUILD Project in Lexical Computing and the Development of the Collins COBUILD English Language Dictionary*, pp. 150–159. HarperCollins Publishers Ltd, London, UK.

Sinclair, J. (1991). *Corpus, Concordance, Collocation*. Oxford University Press, Oxford, UK.

Sinha, R. S., & Mihalcea, R. (2014). Explorations in lexical sample and all-words lexical substitution. *Natural Language Engineering*, *20*(1), 99–129.

Snyder, B., & Barzilay, R. (2007). Multiple Aspect Ranking Using the Good Grief Algorithm. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pp. 300–307 Rochester, New York. Association for Computational Linguistics.

Sprent, P., & Smeeton, N. C. (2007). *Applied Nonparametric Statistical Methods* (4th edition). Texts in Statistical Science. Chapman and Hall/CRC.

Stewart, D. (2010). *Semantic Prosody: A Critical Evaluation*. Routledge, New York, NY, US.

Stone, P. J., Dunphy, D. C., Smith, M. S., & Ogilvie, D. M. (1966). *General Inquirer: A Computer Approach to Content Analysis*. The MIT Press.

Stubbs, M. (2001). *Words and Phrases: Corpus Studies of Lexical Semantics*. Blackwell Publishing, Oxford, UK.

Turney, P. (2002). Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pp. 417–424 Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Wang, T., & Hirst, G. (2010). Near-synonym Lexical Choice in Latent Semantic Space. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pp. 1182–1190 Beijing, China. Coling 2010 Organizing Committee.

Wiebe, J., Wilson, T., Bruce, R., Bell, M., & Martin, M. (2004). Learning subjective language. *Computational Linguistics*, *30*(3), 277–308.

Xu, P., Chelba, C., & Jelinek, F. (2002). A Study on Richer Syntactic Dependencies for Structured Language Modeling. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL'02)*, pp. 191–198.

Yang, Y., & Pedersen, J. O. (1997). A Comparative Study on Feature Selection in Text Categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning*, pp. 412–420 San Francisco, USA. Morgan Kaufmann Publishers Inc.

Yu, H., & Hatzivassiloglou, V. (2003). Towards Answering Opinion Questions: Separating Facts from Opinions and Identifying the Polarity of Opinion Sentences. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing (EMNLP 2003)*, pp. 129–136 Sapporo, Japan.

Yu, L.-C., Shih, H.-M., Lai, Y.-L., Yeh, J.-F., & Wu, C.-H. (2010). Discriminative Training for Near-Synonym Substitution. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pp. 1254–1262 Beijing, China. Coling 2010 Organizing Committee.

Yuret, D. (2007). KU: Word Sense Disambiguation by Substitution. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pp. 207–214 Prague, Czech Republic. Association for Computational Linguistics.