Semantic Knowledge in Word Completion

Jianhua Li
Department of Computer Science
University of Toronto
Toronto, Ontario, Canada M5S 3G4
janeli@cs.toronto.edu

Graeme Hirst
Department of Computer Science
University of Toronto
Toronto, Ontario, Canada M5S 3G4
gh@cs.toronto.edu

ABSTRACT

We propose an integrated approach to interactive word-completion for users with linguistic disabilities in which semantic knowledge combines with *n*-gram probabilities to predict semantically more-appropriate words than *n*-gram methods alone. First, semantic relatives are found for English words, specifically for nouns, and they form the semantic knowledge base. The selection process for these semantically related words is first to rank the pointwise mutual information of co-occurring words in a large corpus and then to identify the semantic relatedness of these words by a Lesk-like filter. Then, the semantic knowledge is used to measure the semantic association of completion candidates with the context. Those that are semantically appropriate to the context are promoted to the top positions in prediction lists due to their high association with context. Experimental results show a performance improvement when using the integrated model for the completion of nouns.

Categories and Subject Descriptors

H.5.2 [**User interfaces**]: Natural language

General Terms

Algorithms, human factors, languages, theory

Keywords

Word completion, linguistic semantics, pointwise mutual information.

1. INTRODUCTION

Word completion, sometimes also known as *word prediction*, is the task of guessing, as accurately as possible, the word that a user is in the process of typing. After the user has typed one or more characters (a *prefix string*), a short list of likely words beginning with those characters is displayed—a *prediction list*; if the intended word is shown, the user may select it with a single keystroke or mouse-click, thereby saving a few keystrokes (see Figure 1). Otherwise, the user continues to type characters until the desired word is predicted or the word has been completely typed. This method has been widely applied to assist physically disabled users for whom every keystroke is an effort, and also as an aid to those with learning or other cognitive disabilities for whom such cues may be helpful. Many commercial word completion software packages, such as CoWriter [2] and WordQ [26], are available. Recently, word-prediction techniques have been extended to resolve ambiguous text input on mobile phone numeric keypads, as in the T9 system [23], and other small keyboards [13].

The challenge in word completion is making the quality of the prediction list as high as possible and excluding implausible or ungrammatical words (such as *uncover* and *uncle* in Figure 1). The software must therefore discriminate among the large number of candidates for most prefix strings in order to choose a short list of likely words for display. The extreme situation is when the user enters the first character; thousands of words starting with that character are all candidates at that point. Currently, most word-completion systems employ statistical models such as word *n*-gram models to predict intended words [4–6, 8]. But while these models capture the co-occurrences of neighbouring words, they are weak in capturing long-distance co-occurrence relations between words. Various systems have attempted to use syntactic information to improve the predictions [4, 11, 17, 18]. For example, Fazly and Hirst [4] added part-of-speech *n*-gram information to the traditional word *n*-gram model. Their experiments showed that this improved prediction accuracy and saved users' keystrokes, but only by a small amount. (See [4] for a brief review of earlier uses of parts of speech and syntactic structure.) Taking into account the limitations of statistical strategies in ungrammatical situations, Wood and Lewis [25] employed a parsing algorithm, Windmill, for word prediction. They assumed that if statistical strategies can discriminate a list of grammatically correct words derived from a syntactic parser at the current point of a sentence, then the prediction outputs will meet the user's needs. They used an augmented Phrase-Structure Rule (PSR) grammar. At each point of constructing a sentence, all potential syntactic constituents are considered and expanded by grammar rules. The words fitting the current syntactic categories are sent to the statistical prediction model, which produces a list of prediction outputs. During the prediction, the sentence is parsed and expanded from left to right.

On the other hand, human language processing also involves semantics. When reading through a text, intuitively people may predict upcoming words by the concepts that have already occurred in the article, no matter how far back the concepts are located [3, 9, 10, 14]. For example, one might predict *patient* if *hospital* has occurred before. The challenge for integrating semantic information into word completion is semantic ambiguity. During the prediction process, a large number of candidates may associate with

ASSETS'05, October 9–12, 2005, Baltimore, Maryland, USA.

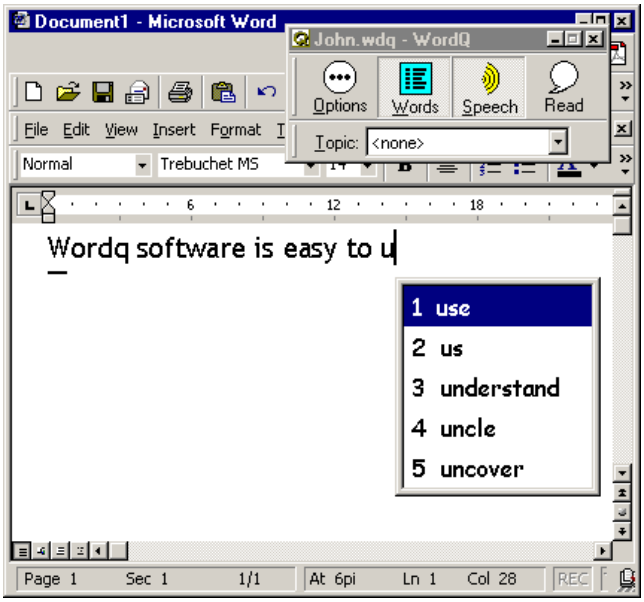

Figure 1: An example of a prediction list in the use of the word completion software WordQ [26].

the previous context in a certain way by a certain sense. Therefore, only models that have a strong disambiguation ability can take advantage of semantics for the completion task; otherwise, semantic information will just be noise that is unhelpful or even deleterious to the process.

Our goal is to take advantage of semantic information in the word completion model. An intuitive way is to measure the semantic association of a prediction candidate with the preceding context and choose the candidates with the strongest association for the prediction list. The method presented in this paper is based on this intuition.

2. CONTRASTING WORK

Our word completion task contrasts with the task of "word prediction" in automatic speech recognition (ASR). In ASR, an acoustic model produces a sequence of words or subwords according to spectral features of sound signals. However, deficits of the model and noisy signals tend to let the model make errors. In order to improve the recognition performance, language models are employed to collaborate with the acoustic model to determine the word sequence for output [12]. There are no human interactions during the whole recognition process. In contrast, in our word completion task the user is, in effect, an oracle who has the authority to decide whether or not a prediction process terminates. If the intended word is in the current prediction list, the user terminates the current prediction by selecting that word from the list and starts the prediction of the next word.

Also different from the word completion we are doing, the so-called "word prediction" task described by Even-Zohar and Roth [3] is to determine a missing word in the context of the text both *before* and *after* the observed position; thus it is not a model of interactive word completion. Moreover, a confusion set containing only two candidates — rather than hundreds or thousands like ours — is used to choose the most likely candidate.

3. INTEGRATING SEMANTICS INTO AN N-GRAM-BASED PREDICTION MODEL

While *n*-gram models can work well with function words, as shown by Fazly and Hirst [4], they are weak in predicting content words that are in function-word–content-word combinations, whereas semantic information can be stronger. Therefore we propose an integrated prediction model in which semantic information is integrated with an *n*-gram model. The two models work as two experts. The final prediction is given by the combination of the two models. Specifically, the predictions of the *n*-gram model are filtered and re-arranged by the semantic model. For those that can be determined by the *n*-gram model as function words, their semantic association with the context is simply regarded as zero and the semantic model is not imposed on them. This separation step helps avoid semantic disturbance on those function words that are favored by the *n*-gram model.

Figure 2 sketches this framework, which subsequent sections will present in more detail. There are two knowledge bases: the *n*-gram knowledge base and the semantic knowledge base. Fazly and Hirst [4] built up the *n*-gram knowledge base and implemented the *n*-gram model. This paper builds up the semantic knowledge base and implements the semantic model. Semantically *related words* and their pointwise mutual information (PMI) are extracted from a large corpus, the British National Corpus World Edition (BNC). Our method of measuring the semantic association of a prediction candidate with the context is based on these related words.

We also propose an algorithm that automatically determines the *salient terms* of a text during the prediction process and uses these terms to measure semantic association for a candidate whenever the candidates find no related words in the context. In addition, the prediction of out-of-vocabulary items — largely named entities — is a problem for *n*-gram models. We employ a "named-entity recorder" to help the prediction of named entities.

4. AN INTEGRATED PREDICTION MODEL

As stated above, our model combines the semantic model with the *n*-gram model. The final prediction outputs are determined by the following formula:

$$\hat{w} = argmax_w(\log P_{ngram}(w) + \log(1 + \lambda \times SA(w, CN))), \quad (1)$$

where $\hat{w}$ is one of the most likely prediction outputs according to the formula (we actually take not a single argmax but the T highest-scoring arguments for a prediction list of size T); the current context CN is a word sequence such as $\ldots, w_{i-3}, w_{i-2}, w_{i-1}$ that the user has already entered in a sentence; $P_{ngram}(w)$ is w's prediction likelihood in the *n*-gram model; $SA(w, CN)$ is the semantic association of w with the context CN; and λ is a parameter used to adjust the weight of semantic association, which has to be determined by experiments on training data. The results to be presented in the later sections were obtained with $\lambda = 10^5$. If w has no semantic relation with current context CN, then SA is 0, and the integrated prediction model is determined by the *n*-gram model alone; otherwise, the *n*-gram information will be used together with semantic association to determine a list of prediction outputs for the intended word. In the algorithm, the prediction candidates for the semantic model come from the output of the *n*-gram model.

Figure 3 presents the prediction algorithm of the integrated model. The variable T in Step 4 is commonly set to 5 or 10. The algorithm covers a single prediction cycle. If the user does not find that the intended word is in the prediction list and instead types a new character, a new cycle begins with the set of candidates reduced accordingly.

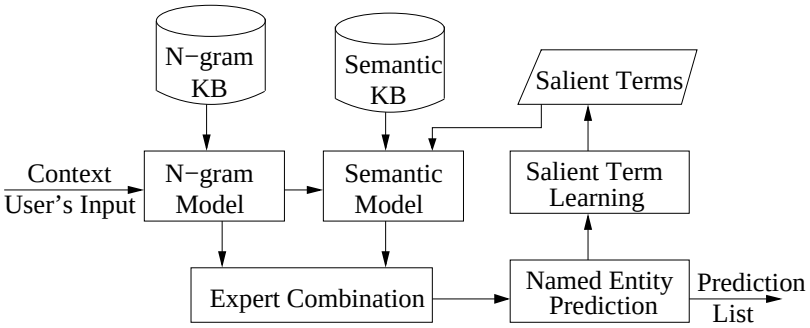

Figure 2: Overview of the integrated word prediction system.

1. The user has entered a prefix string at the current position, say '*sc*' for the word *school*.
2. The *n*-gram model creates a list of prediction candidates for the prefix string.
3. For each candidate w from step 2, compute $\log P_{ngram}(w) + \log(1+\lambda \times SA(w,CN))$.
4. Sort the results by score and output the top T candidates to the user.
5. The user decides whether or not the intended word is in the prediction list.

Figure 3: Prediction algorithm in the integrated model.

5. SEMANTIC ASSOCIATION WITH A CONTEXT

Recently, researchers have employed measures of word relatedness in applications such as word sense disambiguation [1, 20]. However, most current measures that are based on semantic resources such as WordNet are weak in relating words with different parts of speech, such as the relatedness between nouns and verbs, adjectives and nouns, or adverbs and verbs; but these are also crucial for obtaining semantic relatedness with a context. Methods based on co-occurrence (of which *n*-grams are a special case) work across parts of speech, but semantic relatedness is merely induced from the co-occurrence. In this paper, we combine the two ideas: Co-occurrences are filtered by a WordNet-based Lesk-like method, described below, allowing us to consider relatedness both within and across parts of speech.[1] Thus for nouns, we consider not only other nouns but also verbs and adjectives as potential relatives. For verbs, nouns and adverbs may be related.[2] Semantic relatedness is not symmetrical; word w_2 may be a relative of w_1 without w_1 being a relative of w_2.

[1]We do not take the additional step of considering distributional similarity as a proxy for semantic similarity (cf Weeds [24]); see Mohammad and Hirst [19] for a review.

[2]Relatedness to verbs and other parts of speech is not yet implemented at the time of writing; in the evaluation below, the improvements were gained solely with noun relatedness.

5.1 Determining Semantically Related Words

For each word in the vocabulary, we need to determine from a corpus its set of relatives and the degree of relatedness of each relative. This relatedness is used as *SA* in Equation 1. The information is kept in the semantic knowledge base (see Figure 2).

The process of extracting this information from a corpus is illustrated in Figure 4. The corpus used is the British National Corpus, which has part-of-speech tags. We now step through the process.

For each word w whose relatives are to be obtained, the *co-occurring words extractor* finds those words that co-occur with w in a window defined as follows: For co-occurring nouns and verbs, the entire sentence is taken into account, because a sentence is a topic unit and we intuitively expect its nouns and verbs to be conceptually related. On the other hand, the text window for adjectives is more strictly defined: only five words before the target word, including function words and content words. The intuition here is that only the most proximate adjectives relate to the concept properties of the target word. For example, in the sentence *The prospectus gives a report on the students' viewpoint and can be obtained from **individual** offices at some colleges of **higher** education*, the adjectives *individual* and *higher* restrict only the concepts of their most adjacent nouns rather than the other nouns — *higher education* but not (on the basis of this text) *higher prospectus* or *higher offices*.

Similar to Rosenfeld's work [22], where semantically related words are selected by the average mutual information, the *PMI sort processor* computes the pointwise mutual information (PMI) between the words of the pairs of co-occurrences.

$$PMI(w_1,w_2) = \log_2 \frac{P(w_1,w_2)}{P(w_1)\cdot P(w_2)}, \qquad (2)$$

where $P(w_1,w_2)$ is the co-occurrence frequency of the word pair (w_1,w_2) in the corpus, as defined above, and $P(w_i)$ is the occurrence frequency of word w_i in the corpus. Heeding the warning of Manning and Schütze [16] that mutual information has its limits on low-frequency events, we exclude rare words, removing those whose frequency in the BNC is less than a threshold of 50. The co-occurring words are sorted according to their PMI. Those with the highest PMI are automatically regarded as strongly related to the target word. We refer to them as *seed words*, and the exact number chosen is a parameter to the procedure; it is discussed further in the following section. For example, the seed words for *school* include *mathematics*, *parent*, and *teacher*. Words in the remainder of the list are at this stage merely *candidate relatives*, which are sent to the *relatedness filter* for further relatedness identification. For *school*, these include *child*, *program*, and *science*.

In WordNet, the lexicon that we use, each sense of each word (or, more precisely, each set of synonyms) is provided with a *gloss* that

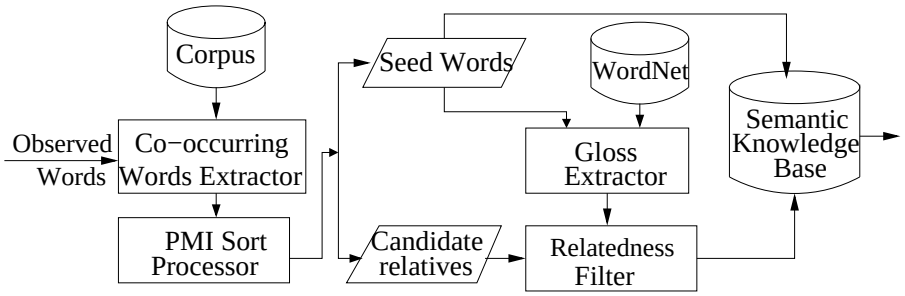

Figure 4: Finding groups of related words.

grammar: studies of the formation of basic linguistic units

parent: a father or mother; one who begets or one who gives birth to or nurtures and raises a **child**; a relative who plays the role of guardian.

mathematics: math, maths, a science (or group of related sciences) dealing with the logic of quantity and shape and arrangement).

teacher: instructor, (a person whose occupation is teaching). a personified abstraction that teaches; "books were his teachers"; "experience is a demanding teacher").

Table 1: WordNet glosses for seed words of *school*.

interprets its meaning and gives some typical examples. Table 1 shows the glosses of some of the seed words of *school*. The words in a gloss tend to be strongly related to the glossed word, such as the word *instructor* in the gloss of *teacher*. These glosses are a good resource to confirm the relatedness of co-occuring words with observed words. Therefore, our relatedness filter uses the WordNet glosses of the seed words to decide whether a candidate is to be considered related. Specifically, a candidate is retained if it occurs in the gloss of any seed word. (If a seed word is in more than one synonym set and hence has more than one gloss, all sets are used.) For example, the candidate word *child* is deemed to be related to *school* because it occurs in the gloss of the seed word *parent* (see Table 1). (We refer to this method as "Lesk-like" because it resembles Lesk's [15] algorithm for word-sense disambiguation, which is based on word overlaps in dictionary definitions.) Table 2 shows some of the final relatives of *school*. Together with each word we store its relatedness to the target word, $Relatedness(w_i, w_j)$:

$$Relatedness(w_i, w_j) = \frac{C(w_i, w_j)}{C(w_i) \cdot C(w_j)}, \quad (3)$$

where $C(w_i, w_j)$ is the count of the number of co-occurrences of the word pair (w_i, w_j) in the corpus, and $C(w_i)$ is the number of occurrences of word w_i in the corpus. So far, the semantic model has been implemented for the prediction of nouns with related nouns and adjectives, and this is what we evaluate below in the evaluation section. We have relationship data for 3031 distinct nouns that occur at least 800 times in 83 million words of the BNC. These nouns include most English common nouns.

5.2 Semantic Association with Context

Given this knowledge base of semantic relatedness, we can compute the semantic association of a prediction candidate with its context just by summing the relatedness of each word pair formed by

Nouns: grammar, governor, curriculum, parent, mathematics, teacher, pupil, liaison, infant, neighbourhood, education, child, ...

Adjectives: secondary, primary, neighbouring, catholic, junior, vocational, compulsory, ...

Table 2: Some nouns and adjectives related to *school*.

Candidate	**Initial rank**	**Related words**	*SA* $\times 10^5$	**New rank**
market	1	*potato, basis*	0.6506	1
media	2	*form*	0.0365	4
marking	3	—	0	5
⋮	⋮	⋮	⋮	⋮
meals	**176**	*potato, form*	**7.2488**	**16**

Table 3: The prediction process after the character *m* is entered in the context of the content words *oats, salads, baked, potatoes, form, basis, daily*.

the candidate with its context words. If

$$CN = \{w_i \mid w_i \text{ is a content word in the sentence}\}$$

is a context and w is a prediction candidate, then the association of w with context CN is computed as follows:

$$SA(w, CN) = \sum_{w_i \in CN} Relatedness(w, w_i). \quad (4)$$

If a context word, say *building*, is not related to a prediction candidate, say *school*, then the value of *Relatedness* is 0. Consequently, if none of the context words relates to a candidate, the candidate will be regarded as having no semantic relation with its context.

6. AN EXAMPLE

Suppose that the user has typed *Oats, salads and baked potatoes form the basis of three daily m*. The *n*-gram model outputs a number of candidates such as *market, media, marking, more, me, my, may, many, must, might, most, man, ..., meals, ...*. Then, the semantic part of the integrated model will measure the semantic association with context for each candidate by Equation 4. Finally, the two parts of information are integrated by Equation 1.

Table 3 illustrates the prediction process after the user types *'m'*. As an example, it lists the situation of the *n*-gram model and semantic association for only the first three candidates and the intended word *meals*. The column labelled *initial rank* shows the candidates' ranks in a candidate list in terms of their *n*-gram probability. The

Cand-idate	Initial rank	Related words	*SA* $\times 10^5$	New rank
men	1	*form, basis*	0.0025	1
members	2	*form*	0.0076	4
means	3	*form*	0.0136	5
⋮	⋮	⋮	⋮	⋮
meals	**39**	*potato, form*	**7.2488**	**2**

Table 4: The prediction process after the character sequence *me* is entered in the context of the content words *oats, salads, baked, potatoes, form, basis, daily*.

column labelled *related words* shows those context words that are candidates' relatives. These context words connect the observed candidate to the context in semantics. The next column shows the value of *SA* in the context, and the last column shows the candidates' new ranks after the combination. In the example, the words *market, media, marking, ...* are at the top of the candidate list from the *n*-gram model, whereas the intended word *meals* is the 176th. Yet *meals* is much more semantically related with the context than other candidates, and the values for semantic association in the table reflect this intuition, i.e., $SA(meals, CN)$ is much higher than that of any other word due to the high relatedness of the word pair ⟨*meals, potato*⟩. Unfortunately, *salad* is not a noun relative of *meals* and so does not contribute to the *SA* of *meals*. *Meals* rises from rank 176 to 16, but this is not enough to get it into the list shown to the user; other words are still more favoured because of high *n*-gram probabilities.

The user therefore needs another keystroke *'e'* to complete the intended word *meals*. Table 4 demonstrates the next prediction cycle. In this process, after the combination, *meals* outperforms almost all other candidates and moves from 39th position to 2nd position, high enough to be included in the list presented to the user. So the model finishes the prediction of *meals* with 2 keystrokes (plus one to indicate acceptance). Since 4 keystrokes are needed for this example with the *n*-gram model alone, we say that the integrated model has saved 2 more keystrokes for *meals* than the *n*-gram model.

7. AUTOMATICALLY LEARNING SALIENT TERMS

Clearly, the semantic part of the integrated model relies highly on the occurrences of related words in the context, e.g., the occurrence of *potato* for *meals* in the sentence of Table 4. If words related to a prediction candidate do not occur in a context, then the semantic part of the model can do nothing to help the prediction. This might be because the candidate is truly unrelated to the context and the candidate is indeed a poor one, or because our extraction of related words as described above was too strict and therefore ignores a large number of words with slightly weaker semantic association. For example, if the number of seed words is set to 5 for *school*, then some clearly associated words, such as *education*, are not extracted. Moreover, the relationships are extracted from the BNC corpus, which cannot cover all language phenomena. Hence a genuinely related prediction candidate might be found to have no relation to the context.

To help counter this, when no semantic associations are found in the present context, we look for *salient terms* — crucial content words — that have been identified up to the current point in the text, and use them as an alternative context to search for semantic associations. For example, the salient terms of an article introducing a patient's medical treatments could be *patient, treatment, therapy, ...*; given the input *Dr. Maurice Slevin, a consultant* and the prefix string *'p'*, there is little semantic information to predict the next word. Nevertheless, if, the previously entered material includes crucial concept terms such as *patient, treatment, therapy, ...*, then the candidate *physician* is more likely to be connected to the material.

In order to make the learning idea practical, two aspects of the words are observed: the word occurrences in the input text and the word frequency in the BNC. Common words such as *life* would not be taken as salient terms in that they usually carry less semantic information than those relatively uncommon words, e.g., *therapy*. A word is deemed to be salient if its frequency in the BNC is less than 15,000 (in 100 million), and it has occurred 6 times or more in the input. These thresholds were determined by experiments on training data. When the number of input occurrences was smaller, say 2 or 3, many terms identified were actually not crucial, and in fact prediction quality was reduced, not improved.

Obviously, this method works better in later parts of a long text, as more salient terms have been learned at that stage; but the user could optionally allow earlier documents on the same topic to be used as well.

Named entities recorded: *Compeyson, Caesar.*

Before NE algorithm	After NE algorithm
China	Compeyson
Cabinet	Caesar
Chiswick	China
Church	Cabinet
⋮	⋮

Table 5: Example of prediction list before and after named entity prediction algorithm for the input *'C'*. The name *Chiswick* happens to appear in the system's *n*-gram model; the names *Caesar* and *Compeyson* do not, but were seen earlier in the text.

8. OUT-OF-VOCABULARY NAMED ENTITIES

Clearly, out-of-vocabulary items (OOVs) are a problem for word completion. In practice, most OOVs will be named entities, and, as with salient terms, are likely to be repeated within the text, so we record all OOVs beginning with a capital letter that the user types, in order to use them in later predictions.[3] During the prediction process, if a word is completed (whether or not it was successfully predicted) and it starts with a capital letter and is not preceded by sentence-end punctuation, then the word is regarded as a named entity and recorded. Then, in predicting, if the first input character is upper-case, recorded named entities starting with the same character are put at the top of the prediction list, ahead of those from the integrated model. An example is shown in Table 5.

[3]The other likely source of OOVs is spelling errors by the user, which we don't want to propagate by suggesting them back to the user; the capital-letter heuristic is a simple way to help prevent this. The present work is part of a larger project that aims to develop spelling and grammar aids for users with cognitive disabilities.

Model	Noun keystrokes (TKS_0)	Spoiled non-noun keystrokes (TKS_1)	Keystrokes needed for nouns (CKS)	Keystrokes needed for spoiled (SKS)	Keystroke saving (%) (KS)
n-gram	22,854	1,454	9,654	393	59
Combin.	22,854	1,454	7,888	654	65

Table 6: Keystroke saving (*KS*) of the integrated model compared with Fazly and Hirst's syntax-and-*n*-gram model on a text with 3700 nouns.

9. SETTING SOME OF THE PARAMETERS

As stated earlier, the contextual association of prediction candidates depends highly on the occurrences of their related words. Salient terms are one way to mitigate this dependence. Two other ways are to increase the number of related words and to extend the observed context.

9.1 The Number of Seed Words

When extracting related words in the earlier section, a crucial factor in the number that are found is the number of seed words permitted. The more seed words there are, the more gloss information will be obtained, and the more words can pass the relatedness filter. That is to say, a larger number of seed words will result in a richer and larger semantic space. On the other hand, a larger space may also create more noise when contributing semantics to the word prediction task — that is, more spurious relationships will be found. To determine an appropriate balance, experiments were carried out and the results will be discussed in the following section.

9.2 The Size of a Prediction Context

Because the semantic model can only use the context before the current word, the length of the context window becomes crucial. A context with more words correspondingly has more chances for prediction candidates to find related words in the context. But again the effects of the semantic model can also be attenuated by a lengthy context in that it will probably lead to more spurious relationships. To observe the effects of context variations, the integrated prediction model was tested by varying the context length from one sentence to four sentences. The following section will present the results.

10. EVALUATION OF THE MODEL

10.1 Keystroke Saving

The traditional evaluation metric of the word prediction task is *keystroke saving* (KS). Keystroke saving reflects what percentage of keystrokes can be saved by the system compared to normal typing of the text.

Since our goal is to explore the contribution of semantic information to content word completion, the integrated model is evaluated in terms of keystroke saving for content words as follows:

$$KS = 1 - \frac{CKS + SKS}{TKS_0 + TKS_1}, \quad (5)$$

Here, *CKS* is the number of keystrokes needed to type content words with the system and *SKS* is the number of keystrokes for those non-content words that actually need *more* keystrokes for completion compared with the *n*-gram model alone, which we call *spoiled words*. For example, if the word *should* could be predicted in some context with one keystroke in the *n*-gram model but requires two keystrokes in the integrated prediction model because semantics initially displaces it with incorrect predictions, then the extra keystroke is a penalty on the model's performance in the formula. In the denominator, TKS_0 and TKS_1 are the number of total keystrokes that would be required to type the content words and the spoiled non-content words without prediction. The presence of *SKS* and TKS_1 reflect how much negative influence the semantic model may bring to the other words.

The training data and the test data are randomly selected from the BNC corpus. These two sets of data are disjoint. The test data contains 3,700 nouns with 22,854 characters in total.

10.2 General Results

The model is evaluated with a simulated user based on that of Fazly and Hirst [4]. Words in a prediction list will be compared with the words in the original text (i.e., intended words). Whenever an original word occurs in the prediction list, the current prediction will be regarded as correct and the number of keystrokes typed so far is recorded for model-performance analysis.

As Garay-Vitoria and Abascal pointed out [7], it is hard to find comparable work, i.e., adding semantic information to improve word completion models, so our baseline for performance is Fazly and Hirst's model [4] in which syntactic information (i.e., part of speech) is combined with word *n*-grams. Table 6 presents a general comparison of the results of the two models. The syntax-and-*n*-gram model achieves a 59% keystroke saving, i.e., only 41% of the possible keystrokes are needed for a user to input nouns. The integrated system obtains a 65% keystroke saving, which is a 14.63% improvement.

This performance improvement suggests that the integrated model does help the traditional *n*-gram model in the completion task; in other words, semantics really contributes to the completion task.

10.3 Varying the Number of Seed Words

We investigated the impact of varying the number of seed words from its initial setting of 50, and hence the number of related words found and the size of the semantic space. The experimental results are listed in Table 7. They demonstrate that varying the number up or down does not enhance or degrade the model performance as we had expected. The change in the number of keystrokes required for content words (*CKS*) is almost exactly balanced by the change in those needed for spoiled words (*SKS*), and *KS* varies only slightly in the third significant figure.

10.4 Varying the Length of a Context Window

We varied the context length for computing *SA* from one sentence to four sentences. Table 8 presents the results. An increase from one to two sentences results in an additional saving of nearly 1%; but the extra improvement with three sentences is slight, and performance starts to drop off again with four sentences. These results indicate that an appropriate length of a context can help the

Number of seed words	Noun keystrokes (TKS_0)	Spoiled non-noun keystrokes (TKS_1)	Keystrokes needed for nouns (CKS)	Keystrokes needed for spoiled (SKS)	Keystroke saving (%) (KS)
10	22,854	709	7,989	319	64.74
30	22,854	1,179	7,905	517	64.96
50	22,854	1,454	7,888	654	64.86
80	22,854	1,684	7,871	746	64.88

Table 7: Keystroke saving (KS) of the integrated model, varying the number of seed words and hence the number of related words found.

Size of context	Keystroke saving (%)
One sentence	64.86
Two sentences	65.64
Three sentences	65.80
Four sentences	65.74

Table 8: Keystroke saving in the integrated model with various context lengths.

model exclude unrelated prediction candidates and save users' efforts.

10.5 Observing the OOV Prediction Strategy

To evaluate the degree to which the OOV prediction strategy assisted the integrated model, we observed the performance both with and without the strategy. Without the strategy, the improvement was only 6.10%, compared to 14.63% with the strategy, as noted earlier. This indicates that the idea of caching recent OOV items is effective and greatly improves the model performance, contributing more than half of the improvement attributable to the complete model.

In fact, this result is not unexpected, for the following reasons. First, it is common that only a limited number of names of people, organizations, or places are involved in an article, and these OOV items are likely to be repeated. Therefore, caching and suggesting these items is very likely to save keystrokes for their following occurrences. Second, OOV items are very often longer than other words. Thus there is a greater potential for keystroke saving if they are predicted early. For example, if the name *Ballantyne* has occurred and been cached in the named-entity recorder, then only one keystroke (plus another for acceptance of the prediction) is needed for its subsequent occurrences, i.e., 8 keystrokes are gained by the OOV strategy. On the other hand, the traditional *n*-gram model is weak in such OOV item predictions and it would probably require all 10 keystrokes to type the name.

11. CONCLUSION

We have proposed a word-completion model based on both *n*-grams and semantic relatedness. A novel Lesk-like relatedness filter is employed in creating the semantic knowledge base that is used to measure the semantic association with a context for prediction candidates. This filter to some extent guarantees that only strongly semantically related candidates can obtain association score and therefore be promoted to the top of prediction lists. The measures of relatedness are rather simple — essentially just mutual information — and our prototype has implemented the method only for nouns. Nonetheless, we were able to improve keystroke saving by 14.63%. The space of possible methods for using semantic information in word completion is large, and our many decisions in the design of our model were almost arbitrary; thus it is likely that further exploration of the space will result in models that have even greater keystroke saving.

So far, our work has focused on the theoretical investigation of semantic knowledge in word completion. Study of its feasibility in practice remains necessary, for keystroke saving is only a crude measure of the quality of a word-completion system [21]. It is possible that users will actually be slowed down by "higher quality" prediction lists, as it might be harder to reject incorrect predictions when they are semantically related; we are presently designing an experiment to test this hypothesis. And for cognitively disabled users, inappropriate predictions may be confusing and hence worse than none at all, so the emphasis of the system must be on rejecting all but the very best predictions. For such users, word completion should be merely part of a larger writing-assistance system, and this is the broader aim of the present project.

ACKNOWLEDGMENTS

We are grateful to Fraser Shein of the Bloorview MacMillan Children's Centre, Toronto, for his continued support and assistance, and to Suzanne Stevenson, Gerald Penn, and Saif Mohammad for comments and suggestions. This work is supported by the Natural Sciences and Engineering Research Council of Canada.

12. REFERENCES

[1] Budanitsky, A. and Hirst, G. Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures. *Workshop on WordNet and other Lexical Resources, Second meeting of the North American Chapter of the Association for Computational Linguistics* (2001), 29–34.

[2] Co:Writer4000. Available at *www.donjohnston.com/catalog/cow4000d.htm*.

[3] Even-Zohar, Y., and Roth, D. A classification approach to word prediction. *Proceedings of the 1st Conference of the North American Chapter of the Association of Computational Linguistics (NAACL)* (2000), 124–131.

[4] Fazly, A. and Hirst, G.. Testing the efficacy of part-of-speech information in word completion. *Proceedings of the Workshop on Language Modeling for Text Entry Methods, 11th Conference of the European Chapter of the Association for Computational Linguistics* (2003), 9–16.

[5] Foster, G., Isabelle, P., and Plamondon, P. Target-text mediated interactive machine translation. *Machine Translation* (1997), 12:175–194.

[6] Foster, G., Langlais, P., and Lapalme, G. User-friendly text prediction for translators. *Proceedings of the conference on Empirical Methods in Natural Language Processing (EMNLP)* (2002), 148–155.

[7] Garay-Vitoria, N. and Abascal, J. A comparison of prediction techniques to enhance the communication rate. In C. Stary and C. Stephanidis (eds.), *User-Centered Interaction Paradigms for Universal Access in the Information Society*, Lecture Notes in Computer Science 3196, Springer-Verlag (2004), 400–417.

[8] Goodman, J., Venolia, G., Steury, K., Parker, C. Language modeling for soft keyboards. *Eighteenth National Conference on Artificial Intelligence* (2002), 419–424.

[9] Hsiao, J.H. A split model to deal with semantic anomalies in the task of word prediction. *Proceedings of the 25th Annual Conference of the Cognitive Science Society* (2003).

[10] Hsiao, J.H. *Dealing with Semantic Anomalies in a Connectionist Network for Word Prediction.* Master's thesis, Simon Fraser University (2002).

[11] Hunnicutt, S. Using syntactic and semantic information in a word prediction aid. *Proceedings of the European Conference on Speech Communication and Technology* (1989), 191–193.

[12] Jeong, M., Kim, B., Lee, G.G. Using higher-level linguistic knowledge for speech recognition error correction in a spoken Q/A dialog. *HLT-NAACL 2004 Workshop on Spoken Language Understanding for Conversational Systems and Higher Level Linguistic Information for Speech Processing* (2004).

[13] Klarlund, N. and Riley, M. Word *n*-grams for cluster keyboards. *Proceedings of the Workshop on Language Modeling for Text Entry Methods, 11th Conference of the European Chapter of the Association for Computational Linguistics* (2003), 51–58.

[14] Kozima, H., and Ito A. A scene-based model of word prediction. Available at *citeseer.ist.psu.edu/97151.html*.

[15] Lesk, M. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from a ice cream cone. *Proceedings, Fifth International Conference on Systems Documentation (SIGDOC '86)*, Toronto (1986), 24–26.

[16] Manning, C., and Schütze, H. *Foundations of Statistical Natural Language Processing.* The MIT Press, Cambridge, Mass. (1999).

[17] McCoy, K.F., Pennington, C.A., and Badman, A.L. A communication aid with artificial intelligence: Communic-ease meets semantic parsing. *Proceedings of Rehabilitation Engineering and Assistive Technology Society of North America (RESNA)'97 20th Annual Conference* (1997).

[18] McCoy, K.F., Pennington, C.A., and Badman, A.L. Compansion: From research prototype to practical integration and alternative communication. *Natural Language Engineering* (1998), 4(1): 73–95.

[19] Mohammad, S. and Hirst, G. Distributional measures as proxies for semantic relatedness. Submitted for publication (2005). Available at *www.cs.toronto.edu/compling/Publications/*

[20] Patwardhan, S., Banerjee, S., and Pedersen, T. Using measures of semantic relatedness for word sense disambiguation. *Proceedings of the 4th International Conference on Intelligent Text Processing and Computational Linguistics* (2003).

[21] Renaud, A. *Diagnostic evaluation measures for improving performance of word prediction systems.* M.Math thesis, University of Waterloo (2002).

[22] Rosenfeld, R. A maximum entropy approach to adaptive statistical language modeling. *Computer, Speech and Language* (1996), 10:187–228.

[23] T9. The easiest way to enter text on a mobile phone. Available at *www.t9.com*.

[24] Weeds, J. *Measures and applications of lexical distributional similarity.* Ph.D. thesis, University of Sussex (2003).

[25] Wood, M.E.J, and Lewis, E. Windmill — The use of a parsing algorithm to produce predictions for disabled persons. *Proceedings of the 1996 Autumn Conference on Speech and Hearing* (1996), 18(9): 315–322.

[26] WordQ. Writing Aid Software. Available at *www.wordq.com*.