Modelling Semantic Knowledge for a Word Completion Task

by

Jianhua Li

A thesis submitted in conformity with the requirements
for the degree of Master of Science
Graduate Department of Computer Science
University of Toronto

Abstract

Modelling Semantic Knowledge for a Word Completion Task

Jianhua Li

Master of Science

Graduate Department of Computer Science

University of Toronto

2006

To assist people with physical disabilities in text entry, we have studied the contribution of semantic knowledge in the word completion task.

We have first constructed a semantic knowledge base (SKB) that stores the semantic association between word pairs. To create the SKB, a novel Lesk-like relatedness filter is employed. On the basis of the SKB, we have proposed an integrated semantics-based word completion model. The model combines the semantic knowledge in the SKB with n-gram probabilities. To deal with potential problems in the model, we propose the strategy of using salient terms and the ad hoc algorithm for the OOV recognition. We tested our model and compared with the model using n-gram probabilities of word and part-of-speech alone and found that our model has achieved significant performance improvement. In addition, test experiments on the algorithm for OOV recognition present a notable enhancement of the system performance.

Dedication

To my parents, Liansheng Li and Yunzhi Zhang

my daughter Xinyi Guo, and my husband Jinfu Guo

For their always love and encouragement. Without their love, life would be meaningless.

Acknowledgements

First and foremost, I would like to thank my supervisor, Professor Graeme Hirst, for his enthusiasm, great insights, and strong support for my academic study. His assistance and support has made this thesis possible. He is an excellent supervisor and I am really pleased to have chance to work with him and get to know him in my life.

I would also like to thank Dr. Fraser Shein for his inspiration and tireless help. He spent quiet an amount of precious time having academic discussions with me. He provided great encouragement and valuable feedbacks on my study.

My special thanks extend to Professor Gerald Penn for his feedbacks and always good suggestions.

I am grateful to Linda Chow for her always administrative assistance.

I would like to take this opportunity to thank my colleagues in the Computational Linguistic group, Afra Alishahi, Fay Baron, Afsaneh Fazly, Diana Inkpen, Saif Mohammad, Jane Morris, Yun Niu, Vivian Tsang, Xiaodan Zhu for their precious time and wonderful discussions.

I would also like to express my gratitude to my friends: Yilan Gu, Xuming He, Hui Lan, Yongmei Liu, Renqiang Min, Naiqi Weng in the Department of Computer Science and a special friend Majorie Kirby for their friendship and encouragement. Also, many thanks to my old friends Lily Jiang, Jane (JiangYan) Reaves, Yuqi Sun, Xiangdong Wei, Huizhen Zhu for their friendship and great help, particularly during my early stay in Toronto.

I am grateful to the financial support from the Natural Sciences and Engineering Research Council of Canada, the Ontario Graduate Scholarship Program, the Department of Computer Science, and the University of Toronto.

Contents

Chapter 1

Introduction

1.1 Alternative and Augmentative Communication

Alternative and Augmentative Communication (AAC) is motivated by needs of helping disabled people communicate with the outside society. It has been partially achieved by designing specialized mechanical devices, customized workstations, and developing assistive computer technologies including word completion techniques (Brown, 1992). Through these technologies, disabled people to some degree can compensate for their physical loss, such as movement impairment and cognitive disabilities.

Assistive computer technologies are directed by various needs of people. Blind people, for example, may demand the assistance of a speech synthesizer so that they can take advantage of their listening ability, while physically disabled people may need special devices or various technologies to make their text entry easy. Clearly, the study of assistive technologies has close relations with the characteristics of various disabilities.

1.2 Word Completion

Word completion, sometimes also known as *word prediction*, is one of the AAC techniques. It 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. Otherwise, the user continues to type characters until the desired word is predicted or the word has been completely typed (Fazly and Hirst, 2003; Li and Hirst, 2005).

Word completion is therefore one kind of word prediction task. Even-Zohar and Roth (2000a,b), Even-Zohar et al. (1999), and Al-Mubaid (2005) described another type of word prediction, where without the user's intervention the complete missing words, rather than characters, are predicted given a context, and the prediction process is not a dynamic one as in the word completion task above. They regarded the prediction of missing words as a classification task, i.e., classifying the intended words from a confusion set of words such as {*make*, *sell*}. Only two words are included in the confusion set, one for the intended word, the other for the confusion word. Compared with *word completion*, the size of prediction candidates here is much smaller and consequently the prediction difficulties are dramatically decreased.

Shieber and Baker (2003) and Foster et al. (2002) described a variant of the word prediction task, where Shieber and Baker (2003) allow the user to use abbreviated compressed forms in order to reduce input characters. The user enters compressed texts, then the system decodes them into full texts. By entering the compressed texts rather than the full texts, the necessary characters for typing to some extent are reduced. Unlike the word completion task, the user does not have to select correct predictions in the process of input. As well, the word prediction task serving in an English-French translator's system is described by Foster et al. (2002), where the system predicts the most likely language strings with an arbitrary length of tokens in the target language for the source language input.

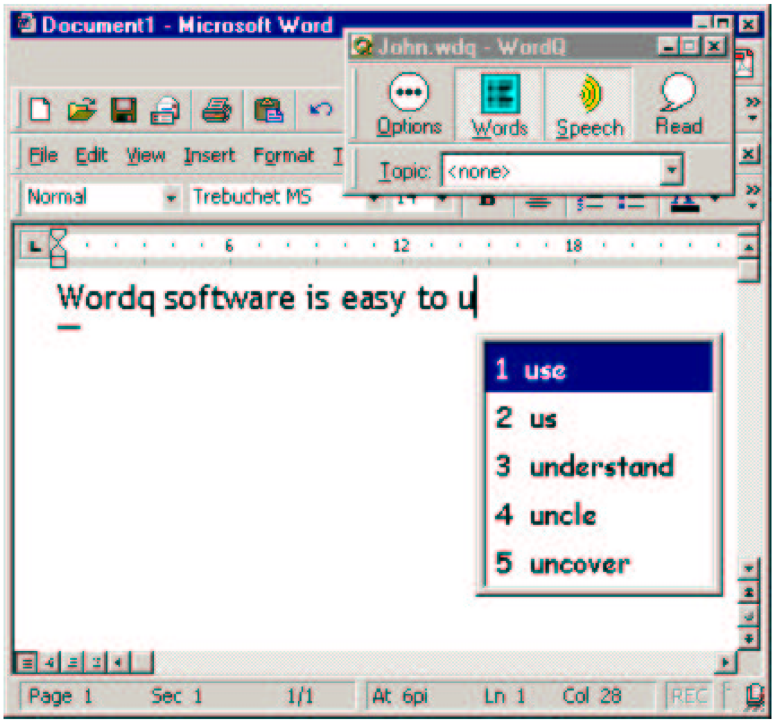
The task of predicting input texts from a Soft Keyboard, a small keypad such as Personal Digital Assistants (PDA) or a mobile phone numerical keypad, is also analogous to the word completion task (Klarlund and Riley, 2003; Venolia et al., 2002; Johansen et al., 2003; Harbusch et al., 2003; Ward et al., 2005; Goodman et al., 2002). Unlike the interactive typing in the word completion task, the user of the small keypad continuously enters characters without stopping and selecting. The system tries to predict the key sequence using a language model and automatically corrects entry mistakes within a word.

Tasks vary in their own characteristics. This thesis concentrates on the problem of the *word completion* task, that is to explore effective strategies to provide most likely predictions for the user at each point of input and save the user's entry effort.

Word completion techniques have 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 difficulties for whom such cues may be helpful. Many commercial word completion software packages, such as WordQ (2004) and Co:Writer (2004), are available for primary schools or health centers. Figure 1.1 and 1.2 are the user interfaces of WordQ and Co:Writer.

Saving physical effort is the first priority of the word completion system for people with physical impairments. It has been well studied by Li and Hirst (2005); Fazly and Hirst (2003); Higginbotham (1992); Carlberger et al. (1997). The amount of effort saved is measured in terms of keystroke savings, i.e., to what extent the necessary keystrokes can be saved in order to complete the text entry. There are various ways to save the user's effort. One is to study effective prediction strategies to reduce keystrokes (Li and Hirst, 2005; Fazly and Hirst, 2003; Carlberger et al., 1997; Magnuson and Hunnicutt, 2002); the other is to design special devices such as reduced keyboards or soft keyboards (Demasco and McCoy, 1992; Venolia et al., 2002; Klarlund and Riley, 2003; Johansen et al., 2003).

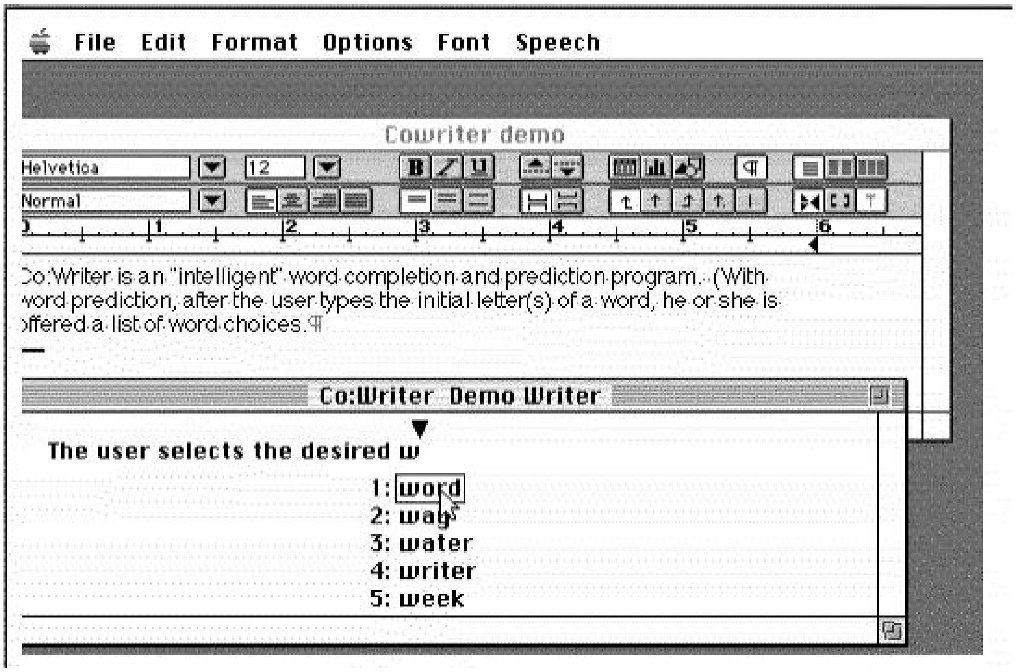
Enhancing typing speed so as to reduce the text composing time is another issue that catches people's attention (Magnuson and Hunnicutt, 2002; Garay-Vitoria and Abascal, 2004; Garay-Vitoria and Gonz, 1997; Koester and Levine, 1994; Card et al., 1980). Experiments show that saving effort does not always mean the reduction of text composing time; in other words, the reduction of keystrokes does not automatically lead to time savings (Magnuson and Hunnicutt, 2002). Studies find that for people with motor dysfunction or really slow typing, keystroke saving means time saving; as for the able-bodied with reasonably typing speed, keystroke


Document1 - Microsoft Word
John.wdq - WordQ
Options
Words
Speech
Read
Topic: <none>
File Edit View Insert Format
Normal
Trebuchet MS
Page 1 Sec 1 1/1 At 6pi Ln 1 Col 28 REC

Wordq software is easy to u

1 use
2 us
3 understand
4 uncle
5 uncover

Figure 1.1: The user interface of the word completion software WordQ (WordQ, 2004).



Co:Writer is an "intelligent" word completion and prediction program. (With word prediction, after the user types the initial letter(s) of a word, he or she is offered a list of word choices.¶

The user selects the desired w

1: word
2: way
3: water
4: writer
5: week

Figure 1.2: The user interface of the word completion software Co:Writer (Co:Writer, 2004).

saving may not be accompanied by the increase of typing speed. Nonetheless, the investigation also shows that in the long run keystroke saving may be more crucial than time saving for both disabled and normal people as it allows them to continue typing for a long period with less fatigue (Bérard and Niemeijer, 2004; Magnuson and Hunnicutt, 2002).

Another important factor, the user's cognitive cost, has been explored when studying the relationship between saving effort and saving time. Cognitive cost refers to the cognitive effort consumed as the user searches the prediction list. During the searching, the user has to syntactically and semantically check the appropriateness of predictions in the prediction list and choose one of them as the intended word. This effort is affected by many internal and external elements such as the user's linguistic knowledge, the interface of the prediction system (a horizontal prediction list or a vertical list and the size of the list, for example), and the content in the prediction list. People with advanced linguistic knowledge can more easily choose the right one from the list than those with less linguistic knowledge, and as a result lower cognitive cost is required. The content of the prediction list refers to how semantically close the predictions are in the list. Whether or not such semantic closeness causes the user's severe confusion and a notable increase of cognitive cost when choosing a word from the list is still not clear in the word completion community.

1.3 Learning Difficulties

Learning difficulties mostly occur among junior students who find it difficult to master language, including reading and writing. They have either a limited vocabulary, or a hard time to find appropriate words, or less skills to command words in their writings as described in (Hyatt and Black, 2005; Goldberg et al., 2003; Handley-More, 2001; Laine, 1998; Laine and Bristow, 1999; Laine, 2000a,b). In particular, Laine and Bristow (1999) described these difficulties as students' written expression problems including *Word Finding, Word Fluency*, and *Word Complexity*.

Compared with physical impairments, learning difficulties are more invisible and consequently attract less attention to ask help from technologies. Whereas, with the successful applications of word completion techniques on physically disabled people in recent decades, more and more studies have been conducted on developing intelligent writing tools for people with learning difficulties.

The study of word completion systems in National Center to Improve Practice in Special Education Through Technology (NCIP) (Word-Prediction, 2005) reported findings in the increase of the students' *vocabulary size*, *writing presentation*, *basic literacy*, *correct spelling*, and *concentration span*. The majority of similar studies showed that word completion techniques allow people with learning difficulties to produce quality written work, to build up self-confidence and independence in writing, and moreover to trigger their enthusiasm about writing.

1.4 Related Work

This section introduces some work in the word completion area. In general, techniques used in word completion can be classified into two categories: statistical-based strategies and frame-based strategies. Statistical methods predict words using linguistic features in statistical models, while the frame-based methods help the prediction by parsing language preference cases including verb constraints.

1.4.1 Statistical Prediction Models

An N-gram-Based Prediction Model

The basic statistical approach to predicting intended words is the *n*-gram model. *N*-gram models estimate the probability of the occurrence of a word at the current position by observing its previous words, and then choose the candidate with the highest probability for that position. For example, suppose the user has typed the word sequence $w_1w_2...w_{n-1}$; the *n*-gram-based word completion model tries to estimate the probability of the next word w_n using the following estimation $P(w_n|w_1,...,w_{n-1})$. That is, the probability of the occurrence of the word w_n is conditioned by the word sequence $w_1w_2...w_{n-1}$.

Since we always encounter texts that we haven't seen before, it is unrealistic to gather all the history information of word sequences to predict the following intended word w_n. To make the estimation realizable, a *Markov assumption* has been made in the *n*-gram model, i.e., only a limited number of previous words may affect the occurrence of w_n and the influence beyond the limit can be ignored. When the previous $n-1$ words are considered, the estimation function is called the *n*-gram model.

The *n*-gram model is one of the important statistical models and employed in most word completion systems described by Cagigas (2001); Carlberger (1998); Fazly and Hirst (2003); Foster et al. (2002); Matiasek et al. (2002). So far, due to the computational complexity of the conditional probability estimation, in practice only one or two previous context words are used, known as the bigram and trigram model.

The Combination of Word N-gram with Other Linguistic Features

The *n*-gram model described above predicts words only based on the occurrences of previous words; no other linguistic features are considered. Nonetheless, syntactic or semantic information obviously plays an important role in the process of prediction as humans do.

The naive but efficient way to take advantage of syntactic information in the word completion task is to integrate part-of-speech tags into the word-based *n*-gram model in order to find syntactically correct outputs (Cagigas, 2001; Carlberger, 1998; Fazly and Hirst, 2003). In most situations, the integration is a linear interpolation. Parsing techniques have also been used in the word completion task (McCoy et al., 1998; Wood and Lewis, 1996). Taking into account the limitations of statistical strategies in ungrammatical situations, Wood and Lewis (1996) employed a parsing algorithm, WINDMILL, for word completion. They assumed that if statistical strategies can discriminate a list of grammatically correct words derived from a syntactic parser at 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. Words fitting the current syntactic categories are sent into the statistical word completion model and the model produces a list of prediction outputs. During the prediction, the sentence is parsed and expanded from left to right. They concluded that syntactic prediction is an effective approach to the word completion task. Their experimental results showed a performance improvement from 50.3% to 55.1% compared with using statistical prediction alone.

Language processing in the human brain is semantics-based, and intuitively semantic evidence would be strong triggers to intended words. Matiasek and Marco (2003) explored the way of adding semantic-trigger-based probabilities into the word-based unigram statistics. The semantic triggers are word collocations residing in a local context window; the occurrence of one word in a collocation pair may trigger the occurrence of the other word. Their work demonstrates that semantic information has a positive effect on the performance of the word completion system though the improvement is not significant.

Kozima and Ito (2004) proposed a scene-based semantics-statistical word completion model where a scene is a sequence of sentences that display a local context. The model predicts words based on the words in the current scene and dynamically detects the scene boundaries after each prediction process. Trnka et al. (2006) explored two possible ways to integrate topic information into the prediction bigram and trigram model. Each topic has a similarity score with the current prediction text and the similarity scores are linearly interpolated with the bigram model. This topic model shows an improvement over the bigram baseline by 1.6% – 1.7%. The other way to use topic information is to compute topic-dependent unigram probabilities, which are then multiplied with the probabilities from a trigram backoff model. This method has a minor improvement over the trigram baseline.

1.4.2 Frame-Based Prediction Models

McCoy et al. (1998) and McCoy and Demasco (1995) proposed a word completion prototype named *Compansion* that predicts sentences by parsing the thematic case frames. The user needs only to input uninflected content words in order and then the system can parse the input, determine each word's part-of-speech and modification relationships between words such as an adjective being a modifier to a noun. There is a semantic parser in the system, which tries to build up a valid semantic representation by filling three main case frames: *case filler preferences*, *case important preferences*, and *higher-order case preferences*. The content of these cases is derived from the analysis of semantic roles of the main verb in the sentence. For example, according to the input *John break window hammer*, the AGEXP (AGent/EXPeriencer) case can be objects in the classes of (*Communication*, *Animate*, *Ergative-Object*) and the THEME can fall in (*Fragile*, *Object*) categories. Each category has corresponding scores indicating their importance to the cases and each case also has its own importance score. By parsing the semantic frames, each word's content is interpreted and the interpretation with the highest score is considered the best prediction of the user's input. Finally, a translator/generator uses the outputs of the semantic parser, automatically adds function words to connect the content words, and generates the sentence that the user intends to input.

Only uninflected content words are needed for input and function words will be automatically added by the system, the frame-based prototype is therefore expected to save the user's effort in terms of keystroke savings.

Chapter 2

Building a Semantic Association Knowledge Base

2.1 Introduction

The challenges of the word completion task are the large number of candidates and the lack of effective strategies to distinguish these candidates. The extreme situation is when the user enters the first character, thousands of words starting with that character all can be the candidates for the current position. N-gram-based statistical models are adept in grasping co-occurrence rules for neighboring words and discriminating prediction candidates by co-occurrence-based statistical probabilities, but weak in capturing co-occurrence relations of long-distance words. For example, in the following sentence:

She didn't think to question the ***treatment*** *her* ***doctors*** *advocated, as a research* ***biochemist*** *who had worked in the* ***pharmaceutical*** *industry for 10 years, the benefits of modern* ***medicine*** *had been instilled in her.*

The word pair (*treatment, doctor*) co-occurs very often and can be easily captured by the N-gram model such as the trigram model. The trigram model can easily predict the occurrence of *doctor* by *treatment*. But statistical models have difficulties to take advantage of co-occurrences of long-distance relations such as *treatment*, *doctor*, *biochemist*, *pharmaceutical* to predict the occurrence of *medicine*. Clearly those words are good helpers for the prediction of *medicine*.

Human beings comprehend language in part on the basis of its semantics. When reading through a text, intuitively people may predict upcoming words by the concepts that have already occurred in the text no matter how far away the concepts are located. For example, one may predict *patient* by the semantic relation if *hospital* occurs before.

Thus, people are investigating various ways to take advantage of semantics in the word completion task such as collocation triggers (Matiasek and Marco, 2003), the Compansion semantic parser system (McCoy et al., 1998), and the scene-based semantic model (Kozima and Ito, 2004). But none of them show exciting results. The difficulty of adding semantics is that the number of prediction candidates which start with the same character sequence as the user's inputs is large and this large number of candidates may all associate with the previous context in a certain way by a certain sense.

In order to investigate effective methods to make use of semantics, we start with building up a semantic knowledge base (SKB), where meanings of words are represented and quantified by other semantically related words. These content words as surface forms of semantics are used to convey the semantics of a context. They will be used to measure how likely a prediction candidate is the user's intended word.

2.2 Corpus Selection

In the SKB, a set of semantically related words that reflect the meaning of an observed content word are extracted from a large-scale corpus. In order to capture the general picture of the meaning of the word, the corpus that is selected for the construction of SKB should contain most language phenomena of the observed word.

The corpus used for building the SKB in this thesis is the British National Corpus World Edition (BNC) with about 100 million words. It collects samples of written modern British

English from a wide range of sources including newspapers, specialist periodicals and journals for all ages and interests, academic books and popular fictions, etc. Texts are chosen for inclusion according to three selection principals: domain (subject field), time (within certain dates), and medium (book, periodical, etc.). The collection of samples therefore is balanced among styles and varieties, not limited to any particular subject field or genre.

BNC Tagging

The BNC corpus is encoded according to the Guidelines of the Text Encoding Initiative (TEI), and is marked up by SGML (Standard Generalized Markup Language) using ISO standard 8879, which separates and marks a variety of structural properties of texts such as headings, paragraphs, lists, etc. The text is annotated by CLAWS (automatic part-of-speech tagger) system, i.e., the parts-of-speech of words and other tokens in the text are explicitly marked as follows:

<s n=”2”><w AT0>The <w NN1>search <w PRP>for <w AT0>an <w NN1>actress <w TO0>to <w VVI>play <w NP0>Scarlett <w NP0>O’Hara <w VVD>took <w CRD>two <w CJC>and <w A T0>a <w DT0>half <w NN2>years <w CJC>and <w CRD>thousands <w PRF>of <w NN2>dollars<c PUN>

Words are tagged with modifiers such as *<w NN1>*, which show the properties of the current token including the token’s class code and its part-of-speech code. For example, *<w NN1> search* designates that *search* is a word and its part-of-speech is a singular noun (*NN1* is one of the part-of-speech codes for singular nouns). The token of *<w CRD> two* tells that *two* is a word of a cardinal number. Table 2.1 gives the BNC basic tagset, known as the C5 tagset.

Tag	**Description**
AJ0	Adjective (general or positive) (e.g. *good, old, beautiful*)
AJC	Comparative adjective (e.g. *better, older*)
AJS	Superlative adjective (e.g. *best, oldest*)
AT0	Article (e.g. *the, a, an, no*)
AV0	General adverb: an adverb not subclassified as AVP or AVQ (see below) (e.g. *often, well, longer (adv.), furthest.*
AVP	Adverb particle (e.g. *up, off, out*)
AVQ	Wh-adverb (e.g. *when, where, how, why, wherever*)
CJC	Coordinating conjunction (e.g. *and, or, but*)
CJS	Subordinating conjunction (e.g. *although, when*)
CJT	The subordinating conjunction *that*
CRD	Cardinal number (e.g. *one, 3, fifty-five, 3609*)
DPS	Possessive determiner-pronoun (e.g. *your, their, his*)
DT0	General determiner-pronoun: i.e. a determiner-pronoun which is not a DTQ or an AT0.
DTQ	Wh-determiner-pronoun (e.g. *which, what, whose, whichever*)
EX0	Existential there, i.e. *there* occurring in the *there is* ... or *there are* ... construction
ITJ	Interjection or other isolate (e.g. *oh, yes, mhm, wow*)
NN0	Common noun, neutral for number (e.g. *aircraft, data, committee*)
NN1	Singular common noun (e.g. *pencil, goose, time, revelation*)
NN2	Plural common noun (e.g. *pencils, geese, times, revelations*)
NP0	Proper noun (e.g. *London, Michael, Mars, IBM*)
ORD	Ordinal numeral (e.g. *first, sixth, 77th, last*).

Continued on next page

Tag	**Description**
PNI	Indefinite pronoun (e.g. *none, everything, one* [as pronoun],*nobody*)
PNP	Personal pronoun (e.g. *I, you, them, ours*)
PNQ	Wh-pronoun (e.g. *who, whoever, whom*)
PNX	Reflexive pronoun (e.g. *myself, yourself, itself, ourselves*)
POS	The possessive or genitive marker ’s or ’
PRF	The preposition *of*
PRP	Preposition (except for *of*) (e.g. *about, at, in, on, on behalf of, with*)
PUL	Punctuation: left bracket, i.e. *(* or *[*
PUN	Punctuation: general separating mark, i.e. . ,*! , : ;* - or *?*
PUQ	Punctuation: quotation mark, i.e. ’ or ”
PUR	Punctuation: right bracket, i.e. *)* or *]*
TO0	Infinitive marker *to*
UNC	Unclassified items which are not appropriately considered as items of the English lexicon.
VBB	The present tense forms of the verb BE, except for *is, ’s: i.e. am, are, ’m, ’re* and *be* [subjunctive or imperative]
VBD	The past tense forms of the verb BE: *was* and *were*
VBG	The -ing form of the verb BE: *being*
VBI	The infinitive form of the verb BE: *be*
VBN	The past participle form of the verb BE: *been*
VBZ	The -s form of the verb BE: *is, ’s*
VDB	The finite base form of the verb BE: *do*
VDD	The past tense form of the verb DO: *did*
VDG	The -ing form of the verb DO: *doing*
VDI	The infinitive form of the verb DO: *do*

Continued on next page

Tag	Description
VDN	The past participle form of the verb DO: *done*
VDZ	The -s form of the verb DO: *does, 's*
VHB	The finite base form of the verb HAVE: *have, 've*
VHD	The past tense form of the verb HAVE: *had, 'd*
VHG	The -ing form of the verb HAVE: *having*
VHI	The infinitive form of the verb HAVE: *have*
VHN	The past participle form of the verb HAVE: *had*
VHZ	The -s form of the verb HAVE: *has, 's*
VM0	Modal auxiliary verb (e.g. *will, would, can, could, 'll, 'd*)
VVB	The finite base form of lexical verbs (e.g. *forget, send, live, return*) [Including the imperative and present subjunctive]
VVD	The past tense form of lexical verbs (e.g. *forgot, sent, lived, returned*)
VVG	The -ing form of lexical verbs (e.g. *forgetting, sending, living, returning*)
VVI	The infinitive form of lexical verbs (e.g. *forget, send, live, return*)
VVN	The past participle form of lexical verbs (e.g. *forgotten, sent, lived, returned*)
VVZ	The -s form of lexical verbs (e.g. *forgets, sends, lives, returns*)
XX0	The negative particle *not* or *n't*
ZZ0	Alphabetical symbols (e.g. *A, a, B, b, c, d*)

Table 2.1: BNC tagset (BNC, 2000)

BNC Domain Information

Texts in the BNC are categorized into nine broad domains as follows in Table 2.2 based on the pattern of book publishing in the UK:

Applied science
Arts
Belief and thought
Commerce and finance
Imaginative
Leisure
Natural and pure science
Social science
World affairs

Table 2.2: BNC domains

Imaginative texts are fictional texts or texts that are generally perceived to be literary or creative. Such texts are hard to be categorized into any of the other domains listed in Table 2.2 as they are not centralized to a certain subject. In comparison with *Imaginative* texts, other texts are called *Informative* texts because their content focuses on one of the specific domains.

2.3 Word Representation in the Knowledge Base

Words exist in a context, and word meaning is determined or disambiguated by the context. Generally, people grasp word meaning through a multi-dimensional picture in which word characteristics are exhibited in various aspects, including word concepts, potential behaviors and properties. For example, the meanings of nouns are conveyed by other nouns in a context; their potential behaviors are presented by neighboring verbs; and neighboring adjectives define their property scope. In the sentence "*Secondary schools are also keen to establish positive* *relationships with neighbouring primary schools so that their school becomes the automatic choice of child and parent for the next phase of education*”, the meaning of *school* in this context is expressed by *child, parent, education*; *school* can have the behavior of *being established*; and *school* can be constrained by *secondary, primary,* and *neighboring* shown in Figure 2.1.

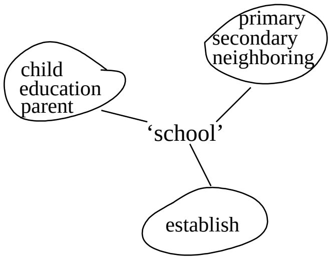


Figure 2.1: The semantic space for *school*.

On the basis of this observation, a set of semantically related words is used to represent various aspects of observed content words. These semantically related words are included in the SKB. In the SKB, nouns have a multi-dimensional semantic space that consists of related nouns slot, verb slot, and adjective slot as shown in Figure 2.2, where the term *slot* refers to a set of nouns, a set of verbs, or a set of adjectives.

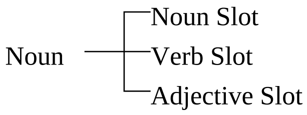


Figure 2.2: The semantic space for nouns.

Similarly, a two-dimensional semantic slot space is built for verbs. As shown in Figure 2.3, a noun slot shows the possible agents or themes of an observed verb, and an adverb slot describes the way or degree of the action conducted by the observed verb.

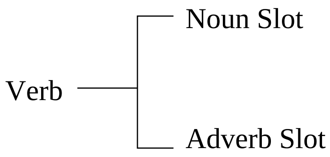


Figure 2.3: Semantic slots for verbs.

Word meaning is conveyed by word concepts, behaviors, and property constraints. These features show co-occurrence characteristics through the words that frequently co-occur with observed words. However, not all of the frequently co-occurrent words are taken as features. For example, only the neighboring, frequently-co-occurrent adjectives are used to present the properties of noun concepts. Section 2.4.3 gives details of constructing the SKB.

These semantically related words are saved in the SKB in a quantitative form that quantifies semantic relationships between the observed word such as *school* and its related words such as *child, parent, education* (see Section 2.4.3). The quantified relation then is integrated into the statistical *n*-gram model to estimate the likelihood for a prediction candidate. The top *n* candidates with the highest estimation values are chosen to output as the prediction list (see Section 3.3.1).

2.4 Semantic Association between Words

Frequent co-occurrences of a pair of words indicate that the two words are related to each other; on the other hand, if two words are related, they will exhibit co-occurrences as a whole in a language. Words in the SKB are related to their observed words since they have co-occurrence characteristics. To extract related words, pointwise mutual information (PMI) is employed. Moreover, to mitigate the disturbance of semantic noise[1], WordNet as a lexical resource is used to help PMI in the following Lesk-like procedure. This section presents how to extract related words from BNC.

In order to guarantee that these extracted words are helpful to the completion task, related words are required to be strongly related to observed words. In other words, only strongly related words are trusted and used to describe the characteristics of observed words. To find a set of semantically related words for an observed word, we extract **strongly related** words

[1] In the word completion task, words with same prefix letters all can be candidates for current positions. But only one is wanted by the user. Thus, candidates except the intended one are all regarded as a kind of linguistic noise that makes the completion task harder.

from BNC corpus, then PMI measurement and lexical resources are employed to confirm the strength of such relationships. Figure 2.4 describes the selection process for semantically related words. The following sections can explain the details of the selection process.

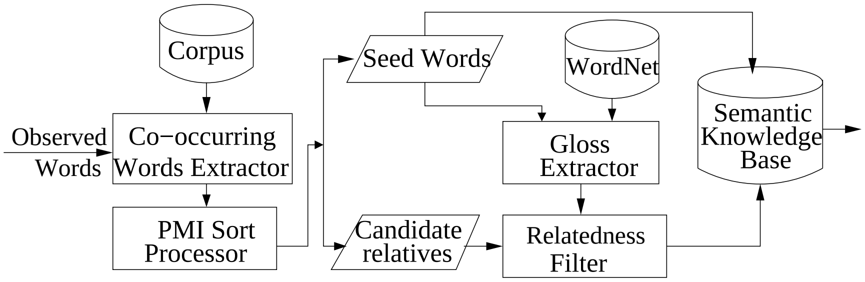


Figure 2.4: The flow chart for extracting related words.

2.4.1 Pointwise Mutual Information

Pointwise mutual information is widely used for discovering interesting collocations in Natural Language Processing. It reflects the amount of information provided by the occurrences of one term x about the occurrences of another term y. Intuitively, if term x and y co-occur very often, then when one sees the occurrence of x, one may naturally expect the occurrence of y in the context. For example, in the sentence *students are writing their essay exam*, the occurrence of *writing* strongly implicates the occurrence of *essay* and *exam* in the future context, and we say the occurrences of x contain a rich amount of information for the occurrences of y; in other words, we expect a high PMI value for such a mutually related word pair.

To reflect and measure such a language intuition, PMI is mathematically defined by Formula 2.1. Suppose $P(x)$ is the occurrence probability of an interesting term x in corpora, $P(y)$ is the occurrence probability of a term y, $P(x,y)$ is the co-occurrence probability of the terms x and y in corpora. Then the PMI is measured as follows:

$$PMI(x,y) = \log_2 \frac{P(x,y)}{P(x)P(y)}. \tag{2.1}$$

Although PMI can be used in most situations to predict the correspondence between two language items such as words, many researchers point out that it has difficulties in describing low-frequency events and does not perform well when encountering sparse data (Church and Gale, 1991; Manning and Schütze, 2001). Say the term x seldom occurs in the corpus, but almost co-occurs with the term y, i.e., $P(x)$ is low but $P(x,y)$ is high. The PMI value of this word pair therefore tends to be high. This limitation of PMI often causes problems in the applications where candidates are ranked by their PMI values because very often candidates with low-frequency remove promising candidates from consideration by their high PMIs.

Similar to the work of Rosenfeld (1996), where semantically related words are selected by the average mutual information, we use the *PMI sort processor* in Figure 2.4 to compute the PMI between the word pairs. Heeding the warning of limits of mutual information on low-frequency events, we exclude rare words by removing those whose frequency in the BNC less than a threshold of 50. Co-occurring words are sorted according to their PMIs. Those with the highest PMI are automatically regarded as strongly related to the target word. They are called *seed words*, and the seed word 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, namely with lower PMI values, 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*.

2.4.2 Lexical Resources

Although PMI indicates that some words are more associated with observed words than the others, it is impossible to only rely on PMI to decide such related words due to the limits of PMI. On the other hand, semantic research has shown that lexical resources are important knowledge bases for acquiring semantic relations.

Lexical resources usually contain lexical information such as word part-of-speech, word frequency, and sense frequency. Some resources also include typical collocations or idiomatic expressions (Jarmasz and Szpakowicz, 2001). Moreover, lexical resources such as WordNet contain semantic information of words including word senses, word-sense definitions, and semantic relationships such as hypernym, hyponym, and synonym between various senses. In addition, FrameNet and VerbNet include subcategorization information of verbs, which describes the potential behaviors of verbs. The linguistic knowledge residing in lexical resources is concluded by linguistics and lexicographers, therefore is the rich historical knowledge of human language. Thus lexical resources are important knowledge resources in natural language processing.

WordNet, as a popular lexical resource, is employed to help PMI measurement identify the strongly related words such as *patient, nurse* to the observed word *hospital* in the construction of the semantic knowledge base.

WordNet

Compared with other resources, WordNet is a richer language resource. Nouns, verbs, adjectives, and adverbs are organized into synonym sets, called *synsets*, that represent underlying word concepts. Various relations between concepts, or synsets, are established. These relationships form the hierarchical structure of WordNet, especially for nouns. There are 79,689 synsets in total for nouns in WordNet 2.0, 13,508 for verbs, 18,563 for adjectives, and 3,664 for adverbs. Many words are polysemous in WordNet; the average polysemy for nouns is 1.23/noun, for verbs 2.17/verb, for adjectives 1.45/adj, and for adverb 1.24/adv. Clearly, verbs are more polysemous than other types of words.

The WordNet hierarchical structure is typically exhibited by the relationships between noun concepts. Table 2.3 shows the relationships of nouns in WordNet.

The hierarchical structure of the other types of content words is not as evident as that of noun concepts, i.e., their hierarchical levels tend to be flat. Table 2.4 designates the concept relationships of verbs, adjectives, and adverbs.

In addition, WordNet provides a *gloss* for each sense of a word (or, more precisely, each synset). The gloss interprets sense meaning and gives some typical examples to help understand the meaning and get to know the usage of the word.

Hyponym
Member holonym
Substance holonym
Part holonym
Member meronym
Substance meronym
Part meronym

Table 2.3: Noun Relationships in WordNet.

Verb	Adjective	Adverb
Antonym	Antonym	Antonym
Hypernym	Similar to	Derived from adjective
Entailment	Participle of verb	
Cause	Pertainym	

Table 2.4: Relationships of verb, adjective, and adverb in WordNet.

Table 2.5 shows the glosses of some of the crucial words for *school*. Since glosses attempt to explain the meaning of the observed word, words used in a gloss tend to be strongly related and less ambiguous to the glossed word, such as the word *instructor* in the gloss of *teacher*. Therefore, these glosses are good resources to confirm the strength of the relatedness of co-occurring words with observed words, i.e., if co-occurring words occur in the gloss of observed words, then the co-occurring words are regarded as strongly related to the observed words.

Word	WordNet gloss
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”).
curriculum	course of study, program, programme, curriculum, syllabus, (an integrated course of academic studies; “he was admitted to a new program at the university”).
governor	the head of a state government, a control that maintains a steady speed in a machine (as by controlling the supply of fuel)

Table 2.5: WordNet glosses for seed words of *school*.

2.4.3 Extracting Relatives from the Corpus

To build up the SKB, PMI and WordNet cooperate together to extract relatives from BNC corpus. Figure 2.4 exhibits the main process of the relative extraction for observed words. First of all, co-occurring words in the context are extracted from BNC corpus, then the *PMI Sort Processor* ranks these words by their PMI values. The top *n* words with the highest PMI are deemed to be *seed words* and the others are considered as *candidate relatives*. After the separation of the extracted words, linguistic knowledge in WordNet plays its role to confirm the semantic relatives of observed words, which will be further sent to the Semantic Knowledge Base. Namely, the *relatedness filter* in Figure 2.4 uses the WordNet glosses of seed words to help decide whether a candidate is to be considered strongly related to the observed word. In other words, 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 more than one gloss, all sets are used). For example, the candidate relative *child* is deemed to be related to *school* because it occurs in the gloss of the seed word *parent* (see Table 2.5). (This method is referred as “Lesk-like” in this thesis because it resembles the algorithm of Lesk (1986) for word-sense disambiguation, which is based on word overlaps in dictionary definitions.)

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

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

Table 2.6: Some nouns and adjectives related to *school*.

Using the above process, we extract relatives for 3031 English nouns. Table 2.6 exhibits some of the relatives of *school*, which reside in the SKB of our completion system.

A Text Window for Extraction

When the semantic knowledge base is being constructed, strongly related words are extracted from a large corpus within a fixed text window. One issue that needs to be considered is the context window, i.e., what would be the context, and what length of the context would be better for the completion task.

Words vary in linguistic functions and are located in various places in a sentence. For example, an adjective acts as a noun descriptor, i.e., to describe properties of a noun. They are expected to occur right before the noun at most of time. Verbs express existence and action; they usually exhibit the possible behaviors of a noun such as *car* being able to *run*, and are expected to appear after the subject noun in a sentence with a subject-verb relationship.

Similarly, adverbs modify verbs and tell how things are done through verb actions. Adverbs can be located theoretically anywhere in a sentence, so they are more flexible compared with adjectives. Clearly, due to linguistic varieties, different types of words require different context windows when extracting strongly related words for the SKB.

To extract strongly related nouns for observed nouns, the smallest text window may be the entire sentence because a sentence is a topic unit and words inside are conceptually related such as in the following sentence, where nouns are conceptually centered on a home care topic.

***Interest** was expressed in all of the organizational aspects of **home care** including **nursing, equipment loans** and the **volunteer programme**.*

On the other hand, the text window for extracting strongly related adjectives for observed nouns is different. Since only those adjectives that occur right before the observed noun can help predict the occurrences of the noun, using the entire sentence as the text window would not be appropriate, instead a short window right before the observed noun is more reasonable. Thus we define as the text window five words before the noun, including function words and content words. 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* only restrict the most adjacent following nouns rather than the other nouns in the sentence, i.e., people can only say *higher education* but not *higher prospectus*.

The verb-object relationship is a good indicator to predict the occurrences of object nouns, the text window for extracting related verbs for observed nouns would be located before nouns. This verb-noun relationship, however, has not yet been investigated in this thesis and so we are not able to answer questions such as if such relationships can greatly contribute to the word completion task and to what extent it can help. But it is indeed an interesting issue that is worthy to explore as intuitively we predict object nouns very often in our daily life according to verbs we have seen before. For example, the sentence *the speaker answered the questions that the audience raised*, where one can easily predict the intended noun *questions* due to the prior occurrence of the verb *answer*.

Semantic Association between Words

The base forms of extracted relatives as well as the measure of the relatedness of these relatives are stored in SKB.

The relatedness of each relative w_i to its observed word w_j is measured as $Relatedness(w_i, w_j)$:

$$Relatedness(w_i, w_j) = \frac{C(w_i, w_j)}{C(w_i) \cdot C(w_j)}, \tag{2.2}$$

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. This measurement of relatedness resembles the measurement of PMI except for the absence of *log* function in Formula 2.1. The following Section 4.7 will discuss the alternative method of measuring semantic relatedness by exact PMI values and the performance evaluation.

So far, only the relatedness of noun-noun and noun-adjective relations has been implemented and saved in the SKB in this thesis. The relatedness of other relationships including noun-verb and verb-adverb relationships is the future work of the completion task. Thus, the evaluation of the word completion model in this thesis focuses only on nouns instead of other types of content words.

2.5 Semantic Association with a Context

The SKB has been built up in previous sections for the word completion task. The SKB contains strongly related words and relatedness measures of word pairs. To predict the next intended word, however, it is useful for us to know the semantic association of each prediction candidate with the previous context. For example, the user has entered the text *the hospital has received a new* and the first character *p*. There are hundreds of prediction candidates starting with *p* and suppose the top three candidates with the highest bigram probabilities are *paper*, *pepper, patient*. Clearly, *patient* is semantically more related to the context and it is a perfect situation that semantic knowledge can help lift the intended word *patient* up to the top of the prediction list. The semantic association of each prediction candidate such as *patient* with the context is such kind of semantic knowledge.

Given the semantic association between words, we compute the semantic association of a prediction candidate with its context by summing the relatedness of each word pair formed by the candidate with its context words. If

$$CN = \{w_i \,|\, w_i \text{ is a content word in the sentence up to this point}\}, \tag{2.3}$$

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

$$SA(w,CN) = \sum_{w_i \in CN} Relatedness(w,w_i). \tag{2.4}$$

If a context word, say *water*, 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, in other words, $SA(w,CN) = 0$.

2.6 Related Work

Semantic relationships of linguistic items including words and word phrases have been well studied and applied to various applications (Kondrak, 2001; Baroni et al., 2002). Budanitsky and Hirst (2001, 2006) evaluated the performance of five measures of semantic relatedness. These measures, proposed by Hirst and St-Onge, Jiang and Conrath, Leacock and Chodorow, Lin, and Resnik, all take advantage of lexical taxonomy of WordNet. They employ the hierarchical structure of WordNet to measure the amount of semantic information commonly shared by a pair of words (Jiang and Conrath, 1997; Leacock and Chodorow, 1998; Resnik, 1995). Budanitsky and Hirst (2001, 2006) compared these measures first by using 65 pairs of words from Rubenstein and Goodenough (1965) and 30 pairs of words from Miller and Charles (1991). They found that the word-pair rating of Jiang-Conrath method is the strongest one correlating with human judgments. They tested various semantic relatedness measures by the application of detecting and correcting real-word spelling errors. The performance comparison on the application shows again that Jiang-Conrath's measure of semantic relatedness outperforms the others in terms of F-measure in general. The experimental results were analyzed and interpreted in detail (Budanitsky and Hirst, 2001, 2006).

Measuring semantic association is an effective way to integrate semantic information with other NLP techniques such as statistical models. Purandare and Pedersen (2004) and Patwardhan et al. (2003) have investigated ways to disambiguate word senses by investigating semantic relatedness measures. Purandare and Pedersen (2004) represented instances of an observed word by the first-order context vectors and the second-order context vectors, which are extracted from raw texts. Twenty neighbouring content words on each side of the word are considered as the context. Word co-occurrences and bigrams are features for the first-order vectors. The space of the first-order vectors is created on training data and used later to discriminate senses of test instances of the observed word by measuring the similarity among vectors. Instead of using raw texts, the semantic relatedness is measured by means of the concept hierarchy of WordNet. They compared the performance of WSD using various semantic relatedness measurements and concluded that employing WordNet gloss overlap and the semantic distance measure of Jiang and Conrath outperform the others.

In addition, Baroni et al. (2002) employed semantic similarity to discover morphologically related words. Instead of using lexical resources, they measured semantic relatedness in terms of mutual information from raw texts. Combining the semantic relatedness with orthographic similarity measured by minimum edit distance, they achieved encouraging results for the morphologically related words. The method of Kondrak (2001) is similar to the work of Purandare and Pedersen (2004), i.e., using WordNet gloss matching to determine the semantic relatedness between a pair of words. Their semantic relatedness measures are used to identify cognates.

They found that the introduction of semantic relatedness dramatically increases the number of identified cognates.

Chapter 3

Integrating Semantics into an N-Gram-Based Prediction Model

3.1 Introduction

More and more linguistic features are integrated into statistical techniques in recent word completion studies, hoping that sophisticated linguistic knowledge can help statistical models provide more accurate and appropriate predictions. Statistical word completion models are therefore developed from the model of word *n*-gram to word-tag (parts-of-speech) *n*-gram (Fazly and Hirst, 2003; Carlberger, 1998), and moreover to the semantic model (Li and Hirst, 2005; Matiasek and Marco, 2003; Trnka et al., 2006).

Word-Sequence Interpolation

Carlberger (1998) introduced the interpolated trigram model for the word completion task, where the bigram model is used as a back-off of the trigram smoother. That is, whenever the occurrence of a word-trigram sequence is zero, the probability of word bigram will play its role as the output of the interpolated model; similarly, the word unigram is used when the occurrence of word bigram sequence is zero. This interpolated trigram model was described by Jelinek and Mercer (1980) with the following form:

$$P(w_3|w_1,w_2) = \lambda_0 f(w_3|w_1,w_2) + \lambda_1 f(w_3|w_2) + \lambda_2 f(w_3), \tag{3.1}$$

where $f(w_3|w_2)$ and $f(w_3)$ are the frequencies of word bigram and word unigram and used to smooth the data in case of sparseness (the zero occurrences of word *n*-gram sequences). The λs are the interpolation parameters that are used to balance the weights among the trigram, bigram, and unigram probabilities and $\lambda_0 + \lambda_1 + \lambda_2 = 1$. The *n*-gram probabilities can be obtained from large-scale training texts.

Word-Tag Interpolation

Fazly and Hirst (2003) explored the effect of incorporating syntactic information with the basic *n*-gram model to predict intended words on the basis of the intuition that syntactic information could reduce the number of prediction candidates that are syntactically incorrect for prediction positions. In their work, the part-of-speech trigrams are interpolated with word bigrams as follows:

$$P(w_i|w_{i-1},t_{i-1},t_{i-2}) = \alpha \times P(w_i|w_{i-1}) + (1-\alpha) \times \max_{t_i \in T(w_i)} [P(w_i|t_i) \times P(t_i|t_{i-1},t_{i-2})], \tag{3.2}$$

where $0 \leq \alpha \leq 1$. The main idea is to first decide a suitable part-of-speech for the current text position, then select as predictions appropriate words that have that part-of-speech. The model parameter α adjusts the significance of the two parts of the model, namely balance the weights between the part-of-speech-trigram model and the word-sequence bigram model. The experimental results show that syntactic information does improve the prediction performance in terms of keystroke saving, but the improvement is small. Thus, it is doubtful whether or not it is worth spending the considerable extra effort such as the response time for the limited amount of performance improvement.

Trnka et al. (2006) investigated the possibility to integrate topic information into language models, where similarity scores between the training topics and the predicting text are calculated and then integrated into the language model. The integrated model is presented as the following:

$$P(w|h) = \sum_{t \in topics} P(t|h) \times P(w|t,h) \tag{3.3}$$

where $P(w|t,h)$ is the probability of the word w given the topic t and the context h. h could be a sequence of words or a sequence of syntactic tags. The estimation of the probability of the topic t is the cosine similarity score of the current context with training texts and calculated by $P(t|h) \approx \frac{S(t,h)}{\Sigma_{t' \in topics} S(t',h)}$. $S(t,h)$ is the cosine similarity score of the topic t with the current context h. Their experimental results show that the topic integration clearly gains a performance improvement over the word-sequence trigram model; however, the increase in terms of keystroke saving is still limited. The best result they obtained is the 1.6% to 1.7% improvement over the bigram baseline.

3.2 Framework

While the work of Fazly and Hirst (2003) shows that the *n*-gram model can work well with function words, semantic information may contribute to the prediction of content words. We propose an integrated word completion model where semantic information is integrated into the *n*-gram model. The two models work as two experts, each in charge of one part of the words: function words and content words. The final predictions are determined by the combination of the two models. Specifically, the predictions of the *n*-gram model are filtered and re-arranged by the semantic model. Figure 3.1 sketches this integrated word completion system.

In Figure 3.1, there are two knowledge bases (KB): the *n*-gram knowledge base and the semantic knowledge base (SKB). Fazly and Hirst (2003) built up the *n*-gram knowledge base and implemented the *n*-gram model. We have built up the SKB and implemented the semantic model. The SKB stores semantically *related words* and their semantic association obtained from a large corpus, the British National Corpus World Edition (BNC). On the basis of *related words* in a context, we propose a method to measure the semantic association of a prediction

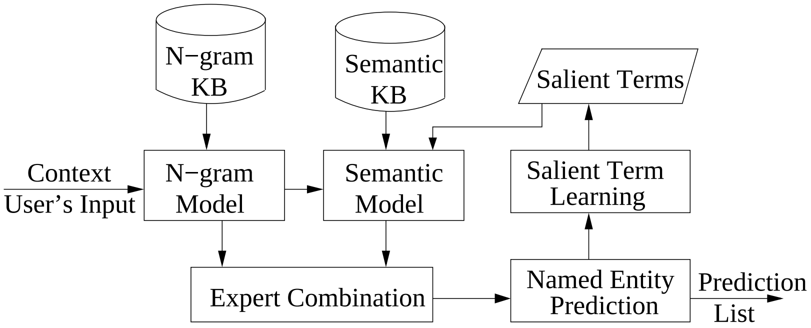


Figure 3.1: An integrated word completion system

candidate (i.e., a candidate for prediction outputs) with the context. This semantic association is combined with *n*-gram probabilities to predict an intended word. In addition, for the possibility that related words are absent from a context, we present the idea of *salient terms*. During the prediction process, the algorithm can automatically learn *salient terms* (defined in Section 3.4) of a text and use these terms to measure the semantic association for a prediction candidate. Besides, the prediction of out-of-vocabulary (OOV) words is a serious problem for *n*-gram models, since OOVs are dramatically variant in texts. In addition, it is impossible to obtain *n*-gram probabilities for all OOVs. Consequently, little help can be expected from *n*-gram models for the OOV prediction. In this thesis, a special OOV prediction strategy is used.

The following sections will present the details of the integrated word completion model.

3.3 An Integrated Prediction Model

Fazly and Hirst (2003) employed a *Tags-and-Words* model for the word completion task, in which a word bigram is combined with a part-of-speech trigram to predict an intended word. Their experiments show that the *n*-gram model works well with the prediction of function words and there is little space to achieve further improvement for these words. Meanwhile, the limits of the *n*-gram model on content words show that a model which can cope with the prediction of content words will probably help the *n*-gram model achieve further improvements.

We propose a semantic association idea to deal with the prediction of content words on the basis of the intuition that people predict content words by context. As shown in Figure 3.1, this semantic model is combined with the *n*-gram model. The final prediction outputs are determined by the following:

$$\hat{w} = argmax_{w_i}(\log P_{ngram}(w_i) + \log(1 + \lambda * SA(w_i, CN))), \tag{3.4}$$

where $\hat{w}$ is one of the most likely prediction outputs according to the equation; the current context *CN* is a word sequence such as $..., w_{i-3}, w_{i-2}, w_{i-1}$ that a user has already entered in a sentence; w_i is one of the prediction candidates; $P_{ngram}(w_i)$ is w_i's prediction likelihood in the *n*-gram model; $SA(w_i, CN)$ is the semantic association of w_i with context *CN*; λ is a parameter used to adjust the weight of semantic association; it has to be determined by the experiments on training data. The model results in Section 4.4 are obtained with $\lambda = 10^5$. If w_i has no semantic relation with current context *CN*, then *SA* is 0, and the integrated word completion 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. The semantic association *SA* has been presented in detail in Section 2.5.

Figure 3.2 presents the prediction algorithm of the integrated model. The variable *T* in Step 4 is commonly set to 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. In the algorithm, the prediction candidates for the semantic model come from the outputs of the *n*-gram model. By observing *n*-gram outputs, it is found that semantic association can only help a range of them enter the prediction list. In other words, the strength of semantic association cannot lift candidates to the top positions of the prediction list if the candidates have really low ranks in the *n*-gram outputs. In practice, 250 *n*-gram outputs are sent to the semantic model to measure the association with context.

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 3.2: Prediction algorithm in the integrated model.

3.3.1 A Prediction 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 2.4. Finally, the two parts of information are integrated by Equation 3.4.

Table 3.1 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 five candidates and the intended word *meals*. The row labeled *n-gram rank* shows the candidates’ ranks in a candidate list in terms of their *n*-gram probabilities. The row labeled *related words* shows those context words that are candidates’ relatives. These context words connect the observed candidate to the context in semantics. The next row shows the value of *SA* in the context, and the last row shows the candidates’ new ranks after the combination. In the example, the words *market, media, marking, more, me, . . .* 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

Original sentence	Oats, salads and baked potatoes form the basis of three daily meals							
Context words	oats, salads, baked, potatoes, form, basis, daily							
Prefix	m							
Candidates	market	media	marking	more	me		meals	
N-gram log probabilities	-2.3069	-2.8199	-2.9960	-3.2067	-3.2332	...	-5.0661	...
N-gram rank	1	2	3	4	5	...	**176**	
Related words	potato, basis	form	NONE	NONE	NONE	...	potato, form	...
Semantic association (SA) ($*10^{-10}$)	0.6506	0.0365	0.00	0.00	0.00	...	**7.2488**	...
$log(1+\lambda*SA)$	0.6286	0.0728	0.00	0.00	0.00		1.5711	
Model combined	-1.6783	-2.7471	-2.9960	-3.2067	-3.2332	...	-3.4950	...
Final rank	1	4	5	8	9	...	**16**	...

Table 3.1: The prediction process after the character *m* is entered

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 favored because of high *n*-gram probabilities.

The user therefore needs another keystroke '*e*' to complete the intended word *meals*. Table 3.2 demonstrates the next prediction cycle. In this process, after the combination, *meals*

Original sentence	Oats, salads and baked potatoes form the basis of three daily meals							
Context words	oats, salads, baked, potatoes, form, basis, daily							
Prefix	me							
Candidates	men	members	means	meeting	member	...	meals	...
N-gram log probabilities	-3.7471	-3.9566	-4.2312	-4.2748	-4.287	...	-5.0661	...
N-gram rank	1	2	3	4	5		**39**	
Related words	form, basis	form	form	basis	form	...	potato, form	...
Semantic association (SA)($*10^{-10}$)	0.0025	0.0076	0.0136	0.0273	0.0076	...	**7.2488**	...
$log(1+\lambda*SA)$	0.0053	0.0162	0.0286	0.0555	0.0162	...	1.5711	...
Model combined	-3.7418	-3.9404	-4.2026	-4.2193	-4.2708	...	-3.4950	...
Final rank	3	5	6	7	8	...	**2**	...

Table 3.2: The prediction process after the character sequence *me* is input

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.

3.4 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 3.2. 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 the 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.

In order to deal with the absence of related words, we propose the idea of salient terms to help prediction candidates obtain context associations. The assumption for the idea is that each article has only one main topic. Sentences and words are expanded and connected around this topic. To support this topic, there exists a set of crucial content words in the article. These words would be likely repeated. They are called *salient terms* of the article.

The salient-term algorithm can automatically learn the salient terms of current input materials. When no semantic association is found in the present context, the prediction model looks for *salient terms* — crucial content words — that have been identified up to the current point in the text and uses them as an alternative context to search for semantic associations. For example, the salient terms of an article that introduces a patient's medical treatments could be *patient, treatment, therapy, ...*; given the input context *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 intended word *physician* is more likely to be conceptually 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 corpus. 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*. In order to filter out common words that have high frequency, a word is deemed to be salient only if its frequency in the BNC is less than 15,000 (in 100 million). Information theory tells that word frequency is negatively proportional to the amount of information that the word possesses. That is, words with lower frequencies have a higher amount of information. However, when the number of input occurrences was small, say 2 or 3, many terms identified are actually not crucial and the prediction performance was reduced rather than improved. In order to guarantee salient terms to be informative, the learning algorithm requires that salient terms at least occur 6 times in the entry context. The thresholds for salient terms were determined by experiments on training data.

Obviously, this method works better in the later stage of the prediction as more salient terms have been learned, but is limited to the prediction with a short context. Nonetheless, the system optionally allows to use earlier documents of the same topic as context. For example, using the word completion system the user can accumulate various texts for his/her work. These texts could be categorized by the topics that the user is interested in. Then during the process of prediction, the texts previous saved and categorized may be taken as contexts and the salient term method works the same way as having a long context.

3.5 Out-of-Vocabulary Named Entities

Out-of-vocabulary items (OOVs) are another problem for word prediction since OOVs are dramatically variant in texts; in practice, most OOVs will be named entities. It is hard to obtain *n*-gram probabilities for all named entities. Consequently, few efforts can be saved by the *n*-gram model for named entities. On the other hand, English named entities are normally longer than other words. Therefore, if we can find an efficient way to predict these names, there would be a large space for the improvement of keystroke saving.

Similar to the assumption for the salient term idea, we assumes that the number of named entities occurring in one article is limited. That is to say, in normal situations only a limited number of people, organizations, or places are involved in an article. Named entities are likely to be repeated. In addition, named entities start with a capital letter. This capital letter actually greatly reduces the candidate space, i.e., only words with the same capital letter are valid candidates. Thus, recording all the named entities that occurred before could be helpful for the prediction of upcoming named entities.

During a prediction process, if a word is completed, no matter whether or not it is successfully predicted, and it starts with a capital letter and is not located at the beginning of a sentence, then the word is regarded as a named entity and recorded. To check whether the current position is at the beginning of a sentence, a set of punctuations (e.g., “. *?* ”) are used as indicators. So far, we do not record the named entities that are located at the beginning of sentences.

When predicting if the first input character is in the capital form, the process of predicting named entities is triggered, which searches recorded named entities starting with the same capital character. Finally, named entities found in the recorded list are put at the top of the prediction list, ahead of those predictions from the integrated model. Table 3.3 presents an example of predicting named entity.

In Table 3.3, through the process of prediction, two words (*Compeyson*, *Caesar*) with the capital letter *C* have occurred. When the user enters a capital letter *C* for the intended word *Compeyson* without using the named entity algorithm, the word completion system outputs predictions as shown in the left column of Table 3.3, where *Compeyson* does not occur in the prediction list and extra keystrokes are needed. On the contrary, when the completion system employs the named entity algorithm, the two capitalized words (*Compeyson, Caesar*) are recorded and arranged at the top of the prediction list as shown in the right column of the table. Words in the previous list are pushed down the list. As a result, the intended named

Named entities recorded: *Compeyson, Caesar.*

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

Table 3.3: Example of prediction list before and after named entity prediction algorithm for the input *‘C’*.

entities are on the prediction list without typing extra keystrokes. The efficacy of this simple strategy has been confirmed through experiments in Section 4.6, where the performance of the word completion system before and after using the named entity algorithm has been compared.

The named entity prediction algorithm works under the assumption that the number of named entities in texts is limited. For situations where this assumption does not exist such as texts of international news, the performance of the algorithm will reduce since too many named entities are cached and it is impossible to simply put them all into the prediction list and moreover the intended one is just one of them.

3.6 Related Work

The word completion task in this thesis 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 (Jeong et al., 2004). There are no human interactions during the whole recognition process. In contrast, in the 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.

Similar to the idea of salient terms introduced in Section 3.4, Seymore et al. (1998) and Iyer and Ostendorf (1996) explored the potential influence of a text topic on the performance of language model. They investigated the topic adaptation for language modeling, i.e., the interpolation of topic information with language models. The work of Seymore et al. (1998) divided content words into three exclusive classes: *on-topic*, *off-topic*, and *general* using statistical tests including *Hotelling's* T^2 *test*, *Kullback-Leibler distance*, χ^2 *test* , and *average mutual information*. The three-way division intends to boost words that are more likely to occur in the texts with the identified topics and suppress words that are unlikely to occur, i.e., increase the probabilities of *on-topic* words and decrease the probabilities of *off-topic* words for identified topics. The topics of articles are manually assigned and topics are possibly overlapped. The three classes of words are predicted through different topic-adapted language models, that is to say, *on-topic* words ($\in V_{ON}$) through the topic-specific language model p_t, *off-topic* and *general* words ($\in V_G$ and $\in V_{OFF}$) through the general language model p_g, which are described as follows:

$$\begin{aligned} w \in V_G &\quad : \quad p(w|h) = p_g(w|h) \\ w \in V_{ON} &\quad : \quad p(w|h) = \lambda_{ON}(h) p_t(w|h) \\ w \in V_{OFF} &\quad : \quad p(w|h) = \lambda_{OFF}(h) p_g(w|h). \end{aligned} \tag{3.5}$$

Here, λ_{ON} and λ_{OFF} are the scale factors of the adapted models for *on-topic* words and *off-topic* words; w, h are the word to be predicted and its context history, respectively. The experimental results showed that the nonlinear topic-adapted language model did decrease the perplexity of the model, i.e.,the performance of the model increases, but surprisingly not as great as they expected compared with linear interpolation.

Distinguished from the above topic adaptation, *on-topic* words, we use salient terms to solve the problem existing in the semantic word completion model in Formula 3.4, namely the lack of occurrences of semantic relatives in a context may weaken the strength of the semantic model. Instead of discriminating topic words in the interpolated *n*-gram model from general words, as in Formula 3.4, topic words are treated equally as other relatives in the context.

Kuhn and de Mori (1990) studied the combination of caching some linguistic items into a language model for speech recognition based on the hypothesis that words used recently are more likely to be used in the near future, i.e., content words "will occur in bursts". Their combined model therefore more emphasizes cached components than the general *n*-gram sequences. Instead of caching only words, they cached the word-POS pair ⟨*word*, *POS*⟩ for each position of a text. The size of the cache is heuristically determined from 5 to 200. When the number of cached items reaches its lower bound, say 5, the combined language model starts to estimate outputs by interpolating the traditional trigram model and the probabilities from cached items, $C_j(W,i)$. $C_j(W,i)$ denotes the probability of cached word W at time i with POS g_j and is calculated from the frequency of W among the N most recent words belonging to POS g_j ($N = 200$ in their experiments). The combined probability is described in Formula 3.6.

$$P(W_i = W | g_i = g_j) = k_{M,j} \times f(W_i = W | g_i = g_j) + k_{C,j} \times C_j(W,i), \tag{3.6}$$

where $k_{M,j}$ and $k_{C,j}$ are model parameters and $k_{M,j} + k_{C,j} = 1$. Comparing with the pure *n*-gram model, their experiments showed a significant performance improvement of the combined model in terms of perplexity. The cached component reflects short-term fluctuations in the frequency of word use.

Chapter 4

Model Evaluation

4.1 Introduction

An informative performance measure should be able to provide objective information for users such that they can choose the model that meets their demands. For example, users with physical impairments would like the word completion models to be evaluated in terms of how much effort they may save provided they use such models; on the other hand, users with linguistic difficulties may prefer an evaluation that reflects how linguistically helpful the word completion model can assist them in their reading and writing. Obviously, word completion models that have a high performance in one aspect may not perform well in other aspects due to various needs of users. Thus, metrics used for the evaluation of completion models need to be able to reflect such various interests and be informative[1].

Before presenting the details of the model evaluation in this paper, some common metrics for word completion are first introduced.

[1] Basically, there are two groups of target users in word completion tasks: people with physical disabilities and people with learning difficulties (Wester, 2003; Magnuson and Hunnicutt, 2002). Some users may show disabilities in both aspects.

4.2 Evaluation Metrics

4.2.1 Effort-Saving Measurement

Keystroke Saving

The traditional evaluation metric for the word completion task is *keystroke saving* (KS). Trying to study the performance of models in terms of execution time, Card et al. (1980) and Embley et al. (1978) first proposed *keystroke saving* as a measure of word completion systems. They found that execution time can be approximated by the time of keystrokes, although ignoring some time factors makes the measurement less accurate. With the assumption that each keystroke costs a constant time unit to finish, *keystroke saving* is used to approximate users' efforts in the process of prediction (Higginbotham, 1992). The more keystrokes that are saved, the better performance a word completion system achieves.

Keystroke saving reflects what percentage of keystrokes can be saved by the system compared to normal typing of the text. It can be calculated as follows:

$$KeystrokeSaving = \frac{K_{total} - K_{typed}}{K_{total}}, \tag{4.1}$$

where K_{total} is the total number of characters contained in the text to be predicted, K_{typed} is the number of keystrokes consumed during the process of prediction. Therefore, the difference between K_{total} and K_{typed} is the keystrokes saved in the process of prediction. The bigger this difference is, the higher performance the word completion system achieves. Keystroke saving therefore becomes a common metric to indicate the degree that the word completion system can save keystrokes, in other words, reduce typing effort for the user if we assume that the number of keystrokes consumed is proportional to the physical effort.

The *keystroke saving* metric has been widely accepted as a measure of evaluation for word completion models both in academic studies that investigate AAC techniques and in commercial companies that produce products for the user with physical disabilities such as WordQ and

Co:Writer. WordQ and Co:Writer are evaluated and compared in terms of *keystroke saving* (Fazly, 2002; Renaud, 2002; Klund, 1995). They are broadly used by school boards and health and special education centers to assist people, especially kids, with easy text entry. The two user interfaces, shown in Figure 1.1 and Figure 1.2, both show that the user only needs to enter the first character for the intended words *use* or *word*; in other words, only one keystroke is needed for both cases rather than three or four keystrokes. *Keystroke saving* as an evaluation measure in such situations is informative to users who are looking for an assistance system that can save their physical efforts.

Hit Rate

Another metric used for the performance measurement is *hit rate*, which represents the percentage of times intended words occur in the prediction list (Fazly, 2002; Fazly and Hirst, 2003). It therefore presents a different aspect of the word completion system. The system having a higher *hit rate* is recognized as having a higher performance. The commercial system *Aurora* prefers to evaluate its system in terms of *hit rate* (Aurora, 2001).

Keystrokes Until Prediction

Keystrokes until prediction measures the average number of keystrokes consumed for completing each word before the word occurs in the prediction list. For example, Fazly (2002) compared the *keystrokes until prediction* for six word completion models and found that the highest value was 2.09 and the lowest one was 1.53. That means the system with the highest value needs 2.09 keystrokes on average to complete one word, while the system with the lowest value requires 1.53 keystrokes. Obviously, the lower this value is, the higher performance the system has. This metric is calculated as following:

$$KeystrokesUntilPrediction = \frac{\sum_{w_i \in T} K_{typed}(w_i)}{Count(T)}, \tag{4.2}$$

where T is the text to be predicted, $K_{typed}(w_i)$ is the number of keystrokes already typed to complete the word w_i, $Count(T)$ is the total number of words in the text.

Accuracy

Accuracy is the percentage of words that can be successfully completed before the last character. The measure of *accuracy* presents the model capacity of completing words regardless of the consumption of keystrokes. Cagigas (2001) described this measure as the prediction coverage, i.e., to evaluate to what degree the test corpus is covered by the word completion system and the user can find his/her intended words from the system. Unlike the *keystroke saving* metric, this measure only cares about the percentage of words that can be predicted with at least 1 character saved. In other words, it observes the capability of the word completion system to provide intended words no matter how many keystrokes the user consumes. As an assistant tool, it is important to always offer intended words without frustrating the user and making him/her lose confidence in continuing to use the system.

4.2.2 Learning Assistance Measurement

Nowadays, word completion techniques are also demanded to provide assistance to people with learning difficulties. For these people, typing efforts are not crucial and thus the measurement of saving efforts is of less interest (Magnuson and Hunnicutt, 2002). To evaluate such word completion models, Renaud (2002) proposed so-called *diagnostic* evaluation measures: *validity* and *appropriateness*. Distinguished from the measures introduced in Section 4.2.1, the two measures identify the actual characteristics of word completion models, namely the syntactic and semantic correctness of the predictions in the list. The hypothesis of this study is that only those predictions that are grammatically correct (for *validity*) and semantically appropriate (for *appropriateness*) should be presented in the prediction list for a well-established word completion model.

Validity

Validity is defined as the proportion of predictions in the prediction list that are grammatically acceptable to the context. It is determined by identifying whether or not a prediction is grammatically consistent with the current position in the context. For example, Figure 1.2 presents a prediction instance and the word completion system Co:Writer provides a list of predictions for the current prediction point. Observing these predictions, we find that all these predictions are syntactically acceptable according to the context, i.e., the *validity* is 100%, while 60% *validity* for WordQ in the shown case in Figure 1.1 due to the presence of *us* and *uncle*. In the work of Renaud (2002), all the predictions presented by the models are taken into account during the evaluation. *Validity* is defined by Formula 4.3,

$$Validity = \frac{the\,number\,of\,grammatically\,consistent\,predictions}{the\,total\,number\,of\,predictions\,the\,system\,presents}. \tag{4.3}$$

A higher value of *validity* implies a better behavior of prediction models.

Appropriateness

Similar to *validity*, *appropriateness* is defined as the proportion of predictions in the prediction list that are semantically plausible for the context, which is more important for the user with linguistic difficulties. This group of users may eagerly want to know to what extent a word completion system could help them with more plausible predictions rather than irrelevant and annoying candidates. Renaud (2002) defined the *appropriateness* as follows:

$$Appropriateness = \frac{the\,number\,of\,semantically\,appropriate\,predictions}{the\,total\,number\,of\,predictions\,presented}. \tag{4.4}$$

The work of judging appropriateness of predictions for the current context was conducted by human judgments (Renaud, 2002).

4.3 Proposed Evaluation Metric

In order to evaluate the performance of word completion models, choosing proper metrics is important as different metrics present different aspects of the models. Answering the following questions may help us decide suitable metrics:

> *Which user group/groups does our word completion model serve?*
>
> *What aspects of the model do we want to observe?*

The main goal of the integrated word completion model proposed in this thesis is to employ semantic knowledge to help traditional *n*-gram models reduce the physical effort for disabled people. Therefore, exploring the contribution of semantic information to word completion is the main interest of the model evaluation. *Keystroke saving* as an effort-related measurement is a proper metric for the evaluation. However, since the semantic knowledge takes effect only on content words, the traditional *keystroke saving* itself defined in Formula 4.1 is not suitable to evaluate the integrated model.

In order to evaluate the integrated model in terms of *keystroke saving* for content words, we propose a modified *keystroke saving* metric presented below:

$$KS = 1 - \frac{CKS + SKS}{TKS_0 + TKS_1}, \tag{4.5}$$

Here, CKS is the number of keystrokes needed to type content words in 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 word completion model because semantics initially displaces it with incorrect predictions, then SKS is 1 and 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. A better word completion model is the one that can greatly reduce the number of keystrokes of content words but not cause dramatic increases of keystrokes of non-content words.

4.4 Experimental Results

Since all words could be prediction candidates as long as they start with a given character, I have built up the semantic knowledge base for 3031 distinct nouns that occur at least 800 times in the BNC corpus. These nouns include most English common nouns. Related words are extracted from the BNC corpus with 83 million words. The corpus has been tagged by the CLAWS system. The training data (3,693 words of which 782 are nouns) and the test data (17,496 words of which 3,700 are nouns) are randomly selected from the corpus. These two sets of data are disjoint.

General results

The model is evaluated with a simulated user based on that of Fazly and Hirst (2003). 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.

The test data contains 3,700 nouns with 22,854 characters in total. Because it is hard to find comparable work, i.e., adding semantic information to improve word completion models, my baseline for performance is Fazly and Hirst's model (2003) in which syntactic information (i.e., part of speech) is combined with word *n*-grams. Table 4.1 presents a general comparison of the results of the two models. The syntax-and-*n*-gram model achieves 0.59 keystroke saving, i.e., only 41% of the possible keystrokes are needed for a user to input nouns. The integrated system obtains 0.65 keystroke saving, which is a 14.63% improvement. The improvement

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	0.59
Combin.	22,854	1,454	7,888	654	0.65

Table 4.1: Keystroke saving ($\boldsymbol{KS}$) of the integrated model compared with Fazly and Hirst's syntax-and-$\boldsymbol{n}$-gram model on a text with 3700 nouns.

of keystroke saving is calculated by the difference of the keystroke saving in the two models divided by 1 minus the keystroke saving of the n-gram model.

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.

4.5 Setting Some of the Parameters

As stated in Section 2.5, the contextual association of prediction candidates highly depends on the occurrences of their related words. If there exist context words which are the related words of the current candidate, then the candidate would have semantic association with the context; otherwise the semantic association would be 0. 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.

4.5.1 Varying the Number of Seed Words

When extracting related words in the section of determining semantically related words, a crucial factor is the number of seed words permitted. The more seed words there are, the more

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	0.6474
30	22,854	1,179	7,905	517	0.6496
50	22,854	1,454	7,888	654	0.6486
80	22,854	1,684	7,871	746	0.6488

Table 4.2: Keystroke saving (***KS***) of the integrated model, varying the number of seed words and hence the number of related words found.

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 completion task — that is, more spurious relationships will be found. To determine an appropriate balance, experiments were carried out.

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 4.2. 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.

Size of context	Keystroke saving
One sentence	0.6486
Two sentences	0.6564
Three sentences	0.6580
Four sentences	0.6574

Table 4.3: Keystroke saving in the integrated model with various context lengths.

4.5.2 Varying the Length of a Context Window

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 word completion model was tested by varying the context length from one sentence to four sentences.

We varied the context length for computing *SA* from one sentence to four sentences. Table 4.3 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 model exclude unrelated prediction candidates and save users' efforts.

4.6 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. The results, shown in Table 4.4, indicate 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:

- 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.

- Very often, OOV items are 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.

Strategy	**%Improvement**
Without OOV	6.10
With OOV	14.63

Table 4.4: Model improvement before and after using OOV prediction strategy.

4.7 Exploring Different Combination Methods

Another method to define relatedness between words is to directly use pointwise mutual information of word pairs. Then the semantic association with context, $SA(w, CN)$, is combined with the syntax-word-based *n*-gram model as follows:

$$\hat{w} = argmax_w(\lambda \log P_{ngram}(w) + (1-\lambda)SA(w, CN)), \tag{4.6}$$

where λ is a parameter to adjust the weight between the *n*-gram model and the semantic association; $SA(w,CN)$ is still the accumulation of $Relatedness(w_i,w_j)$, but the $Relatedness(w_i,w_j)$ now is the PMI between w_i and w_j instead of Equation 2.2.

We compared this alternative method (referred as Method 2 in the following tables) with my earlier integrated model (referred as Method 1). Table 4.5 shows the performance improvements of the two models over the *n*-gram model. The alternative approach obtains this result when λ is 0.85 and the length of a prediction context is one sentence.

Strategy	**%Improvement**
Method 1	14.63
Method 2	14.68

Table 4.5: Model improvements under different combination strategies.

In fact, it is not surprising that the two approaches finally obtain roughly the same results because they use the same linguistic feature — word occurrences — to define relatedness between words. The definition of relatedness in Method 1 does not take a logarithm, instead it is accumulated to calculate *SA* and then the logarithm is taken in the final combination step; on the contrary, this section takes PMI as relatedness (i.e. doing logarithm first) and then accumulates and combines with *n*-gram probabilities. So basically these two methods mathematically have no large differences.

Chapter 5

Conclusions and Future Work

5.1 Conclusion

The goal of this thesis was to explore the contribution of semantic knowledge to the word completion task, in other words, to investigate ways of using the semantic knowledge and whether or not the semantic information can enhance the performance of the word completion system in terms of keystroke saving for people with physical disabilities.

Driven by the goal, we have first constructed a semantic knowledge base (SKB) that stores the semantic association between word pairs such as noun-noun pairs and noun-adjective pairs. This knowledge base is used to measure the semantic association of a prediction candidate with the previous context. On the basis of the SKB, we have proposed an integrated prediction model where the semantic knowledge is combined with *n*-gram probabilities.

In order to deal with the absence of related words in the context, we propose the idea of using salient terms that can help prediction candidates obtain context association. In addition, we present the strategy for the completion of Out-of-Vocabulary named entities, since OOV terms are hard to predict.

In order to explore other potential ways of integrating the semantic knowledge, we have also investigated an alternative approach of combining semantic information, namely pointwise mutual information.

5.2 Contributions

Constructing Semantic Knowledge Base

We have constructed the SKB for the integrated semantics-based word completion model. The semantic relationships between words are extracted from the large-scale corpus, the BNC corpus, and represented by measuring the semantic association between word pairs. In order to create the SKB, a novel Lesk-like relatedness filter is employed. The filter to some extent guarantees that only strongly semantically related candidates can obtain an association score and therefore be promoted to the top of prediction lists. The measures of relatedness are rather simple — essentially equivalent to pointwise mutual information — and the prototype has implemented the method only for nouns. Nonetheless, it was able to improve keystroke saving by 14.63%.

Using Salient Terms

In the integrated semantics-based model, the lack of relatives of the prediction candidates in the context reduces the effect of the semantic information on the prediction results, because not finding relatives in the context is regarded as no semantic relationship with the context; the association score is 0. Therefore, we propose and have implemented the *salient term* idea and use *salient terms* as a kind of context. The presence of salient terms greatly mitigates the problem of lacking relatives and allows the integrated model to gain more keystroke savings.

Predicting OOV Named Entities

Having realized the difficulties of predicting OOV terms, we have proposed and implemented the ad hoc algorithm for the OOV named entities. Due to the specialty of the word completion task, i.e., the prediction candidates have the same prefix sequence as the user's input, the algorithm caches all the named entities and selects the most likely ones as the prediction outputs. Experiments show that this simple strategy significantly enhances the model performance.

Integrating Semantics

Integrating semantics into the word completion model is an important way to promote semantically appropriate predictions to prediction lists such that prediction outputs are not only syntactically correct, but also semantically appropriate. Semantics may be represented in various ways such as semantic association and semantic case frames. We have constructed the SKB in the thesis. On the basis of the SKB, we have implemented the linear integration model. Experiments demonstrate that the integrated information successfully promotes semantically appropriate predictions to the top of the lists and significantly enhances the model performance.

Exploring Alternatives

Moreover, we have investigated the alternatives of integrating semantics into the word completion model. The main difference between the alternative from the integrated model presented in Chapter 3 is the representation of semantic information and the way of combining with *n*-gram probabilities. We found that the two approaches obtain roughly the same results. Section 4.7 analyzes the results and explains reasons.

5.3 Future Work: The Evaluation of Prediction Outputs

Helping people with physical impairments is only one goal of word completion techniques; assisting people with learning difficulties is another one. In order to assist people in their linguistic demands, measuring the quality of prediction outputs would be the next step to move on for the future work. In this section we have reviewed the word-completion work that helps people with linguistic difficulties. The literature review shows that prediction lists that contain many semantically appropriate predictions may be more needed by the users than the lists having fewer such predictions, because such lists can expand users' vocabulary and liberate them from the frustration of not being able to find the words they want. In addition, we will also discuss certain crucial aspects of the evaluation measurement.

5.3.1 Motivation

Keystroke saving is a crude measure of one aspect of a prediction system; it is closely related to the reduction of users' efforts (Carlberger, 1998; Magnuson and Hunnicutt, 2002) and crucial to people with physical disabilities. On the other hand, reducing efforts may not be the main concern of the user with linguistic demands. They would need an evaluation measure that can direct them to find a linguistically helpful system.

Laine and Bristow (1999) considered word finding, word fluency (through word count), and word complexity (variety of words used) as critical elements to effective written expression and productivity — some aspects related to the quality of written products. Word finding and word fluency observe one's ability to find semantically consistent words from memory during a writing process, and word complexity reflects the ability to command word variations. These elements are closely related to the linguistic characteristics of prediction outputs, namely whether or not prediction lists can trigger users' memory or even thoughts and assist them to find appropriate words and moreover to provide more options including some word variations.

5.3.2 The Measure for Prediction Outputs

Quite an amount of work has already been done on the relation between learning difficulties and word completion techniques (Carlberger et al., 1997; Matiasek et al., 2002; Goldberg et al., 2003; Higginbotham, 1992; Hyatt and Black, 2005; Laine and Follansbee, 1994; Laine and Bristow, 1999; Laine, 2000a,b; Matiasek and Marco, 2003; Handley-More, 2001; Word-Prediction, 2005; Zhang et al., 1995). The investigations have been conducted from various perspectives including the state-of-the-art of prediction techniques, suggestions to therapists, and users' responses in government-funded projects that took place over a large number of school districts. General findings demonstrate that students are "not only more engaged and motivated in their writing, but they produce written work that is of greater length and high quality" (Goldberg et al., 2003).

The Learning Disabilities and Technology project presented by Hyatt and Black (2005) proceeded from 1997 through 2002 across 34 school districts in Washington State. The project was administered by the Special Education Technology Center (SETC) at Central Washington University in Ellensburg and evaluated by RMC Research Corporation in Portland, Oregon. The formal evaluation lasted 4 years. Most students were in Grade 3–8. The project uses the word prediction software Co:Writer to help students' writing. Co:Writer uses a topic-related vocabulary. They have observed that students are more willing to write under the help of Co:Writer; students create better written products, and show more confidence in writing.

The Centre for Communicative and Cognitive Disabilities (CCCD) is a Canadian university-based centre at University of Western Ontario. It was established in 1985 and has conducted many research projects closely related to assistive technologies and writing process like Computer Assisted Writing Process Model in Inclusive Settings, for example. In their projects, students between Grade 5 and Grade 8 with learning difficulties use a word prediction tool, **WriteAway**. Twenty-three observed students were required to use **WriteAway** over a twelve-week period (Laine, 2000b). When students enter the first few characters, **WriteAway** generates word lists for them. The word lists are content-specific because the project believes that maintaining a flow of ideas is crucial to the effective use of language and qualitative written expression (Laine and Bristow, 1999). They analyzed the written expression and productivity in students' written products by observing their abilities in word finding, word fluency, and word complexity. They performed a t-test on the number of words used, the variety of words written, and the number of spelling errors (Laine and Follansbee, 1994). The results show that word finding and word fluency of all students have been significantly increased compared with paper-and-pen writing; there is an increase in word variety but not significant; the number of spelling errors decreases; students engage a greater length of time in computers for writing; they are interested in the words in prediction lists and actively search the lists for the words they intend to use. They found that words in the lists cue students to what word they might want to use. However, the major concern in the project is that the effective use of a word prediction system requires a training program, especially at the beginning, to make students familiar with the system. This is confirmed by Hyatt and Black (2005) and Magnuson and Hunnicutt (2002).

In addition, Carlberger et al. (1997), Matiasek et al. (2002), and Matiasek and Marco (2003) presented the attempts of promoting coherent thinking in the writing process by excluding semantically incongruous words from word lists. Carlberger et al. (1997) clearly stated that the integration of semantics is not driven by further saving keystrokes, but motivated by providing more content cue words for the writing. They established four semantic categories for nouns and adjectives. These semantic categories are added to the unigram model. Zhang et al. (1995) examined the impact of the word prediction tool, **ROBO-Writer**, on the written work of students with learning difficulties. Thirty-three children from grades 2, 3, 4, and 5 were tested. They found that children using **ROBO-Writer** produced higher quality written products and fewer spelling errors; moreover, prediction lists of **ROBO-Writer** assisted students in spelling difficult words. This result was confirmed by Magnuson and Hunnicutt (2002) who found most of the time students prefer to accept the predictions of longer content words but would like to type short function words. In order to provide congruous words, Lesher and Rinkus (2005) investigated the method of dynamically switching the prediction model to an appropriate domain. However, this work did not explicitly show whether or not the inflected forms of words shown in one list would cause more confusion to users in decoding which form may be intended and grammatically correct for the context.

Clearly, the above work shows that word prediction systems can significantly assist people with learning difficulties to improve their writing expression — one aspect of qualitative writing. Moreover, the project in CCCD shows that providing word lists with more semantically appropriate content words is an effective way to improve writing expression.

To measure the prediction outputs, some important factors need to be considered in the new measure.

- **The number of appropriate outputs.** The previous literature review shows that word lists that provide more semantically appropriate predictions are capable of assisting people with learning difficulties. The proportion of appropriate prediction outputs in word lists could reflect to what extent the outputs can assist users in word finding, word fluency, and word variations. Ideally, all predictions in the lists should be grammatically correct and semantically appropriate.

- **The degree of appropriateness.** Renaud (2002) proposed the *appropriateness* as a measurement of prediction lists, it is a binary measure, i.e., each prediction output is either appropriate or inappropriate. However, very often, people could not give such clear yes/no answers when they are asked to judge an output given the context. Therefore, using more than two judgment levels for *appropriateness* degree may be helpful for the user.

- **The position of outputs.** The positions of prediction outputs in prediction lists are also important to a prediction system. Experiments in the area of Human-Computer-Interaction found that one can distinguish at most three to five items in a menu at a glimpse (Wester, 2003); namely, predictions at the head of the lists can be easier found than the following ones. Therefore, a prediction system that could promote appropriate predictions to the top of the prediction lists would be preferred by users.

5.3.3 Some Issues in the Experiments

To evaluate the prediction outputs of a semantics-based word completion system, some experiments using human judgments would be necessary.

Human judgments can be influenced by many internal and external factors, such as subjects' language ability, test materials, test approaches, and test duration. Therefore, we need to be really cautious in setting up the experiments. To make experimental results unbiased to the influential factors, the following aspects need to be carefully considered:

a. **Subject selection and the number of subjects.** Subjects are crucial to the experiments. The selected subjects should be representative to the characteristics of target users who have word-finding and word-fluency difficulties.

b. **Test materials.** Users with learning difficulties have limits to express their thoughts by operating language in their written products. Therefore, the linguistic level of test texts for human judgments should be carefully considered. Copestake and Flickinger (1999) pointed out that using logged data is a good way to simulate word prediction algorithms and quickly compare them on realistic data. As well, Magnuson and Hunnicutt (2002) and Wester (2003) did human judgments on users' private logged data.

d. **Test format.** The experiments could be proceeded either by showing prediction outputs on screen and asking users to judge, or showing the outputs on paper. The content is exactly the same. Obviously, paper-based test format is simpler and easy to conduct.

c. **Pilot study.** A pilot study that aims to examine the settings of the experiments and the interactions among the factors involved may be needed. Some preliminary results may be obtained in this phase. The experience in the pilot study may lead the following experiments to be smoothly carried out.

c. **Spectrum.** During the pilot study, the discrete levels for *appropriateness* could be binary; that is, subjects judge the outputs to be either appropriate or inappropriate. With the progress of experiments, like the human evaluation in machine translation tasks described by Sparck Jones and Galliers (1996), more than binary levels such as *"appropriate"*, *"probably appropriate"*, and *"inappropriate"* may be considered.

e. **Test duration.** Psychologically, people are likely to become fatigued when they are required to concentrate on a task for a long time, and then they are likely to make errors. Therefore, like human-judgment-based experiments conducted by Tsandilas and Schraefel (2005) and Wigdor and Balakrishnan (2004), one session of the experiments may last about two hours including a rest break.

f. **Ethical issues.** Very often, human-related experiments have an ethical issue to consider. Subjects in the experiments need to be informed about the purpose of the test, in which form the test will be carried out, are there further impacts on subjects after the test.

Bibliography

Al-Mubaid, Hisham. 2005. Context-based word prediction and classification. *citeseer.ist.psu.edu/729008.html* .

Aurora. 2001. Aurora Systems Inc. Aurora Prediction Software. *http://www.aurora-systems.com* .

Baroni, Marco, Johannes Matiasek, and Harald Trost. 2002. Unsupervised discovery of morphologically related words based on orthographic and semantic similarity. In *Proceedings of the Workshop on Morphological and Phonological Learning of ACL*, 48–57.

Bérard, Christian, and David Niemeijer. 2004. Evaluating effort reduction through different word prediction systems. In *Proceedings of the IEEE International Conference on Systems, Man & Cybernetics: The Hague, Netherlands*, 2658–2663.

BNC. 2000. BNC2 POS-tagging manual. *http://www.comp.lancs.ac.uk/ucrel/bnc2/bnc2guide.htm#tagset* .

Brown, Carl. 1992. Assistive technology computers and persons with disabilities. In *Communication of the ACM*, 36–45.

Budanitsky, Alexander, and Graeme Hirst. 2001. Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures. In *Workshop on WordNet and Other Lexical Resources, Second meeting of the North American Chapter of the Association for Computational Linguistics*. Pittsburgh.

Budanitsky, Alexander, and Graeme Hirst. 2006. Evaluating wordnet-based measures of semantic distance. *Computational Linguistics* 32:13–47.

Cagigas, Sira E. Palazuelos. 2001. *Contribution to word prediction in Spanish and its integration in technical aids for people with physical disabilities*. PhD. thesis, Laboratorio de Tecnologías de Rehabilitación, Dpto. de Ingeniería Electrónica, Universidad Politécnica de Madrid.

Card, Stuart K., Thomas P. Moran, and Allen Newell. 1980. The keystroke-level model for user performance time with interactive systems. In *Communications of the ACM, 23(7)*, 396–410.

Carlberger, Alice, Johan Carlberger, Tina Magnuson, M. Sharon Hunnicutt, Sira E. Palazuelos-Cagigas, and Santiago Aguilera Navarro. 1997. Profet, a new generation of word prediction: an evaluation study. In *Workshop on Natural Language Processing for Communication Aids of the Association of Computational Linguistics*.

Carlberger, Johan. 1998. *Design and implementation of a probabilistic word prediction program*. Master's thesis. The Royal Institute of Technology in Stockholm, Sweden.

Church, Kenneth W., and Willian A. Gale. 1991. Concordances for parallel text. In *the Seventh Annual Conference of the UW Centre for the New OED and Text Research*, 40–62.

Copestake, Ann, and Dan Flickinger. 1999. User evaluation of a word prediction system. In *Augmentative and Alternative Communication: new directions in research and practice*. Filip Loncke, John Clibbens, Helen Arvidson and Lyle Lloyd (ed.).

Co:Writer. 2004. Co:writer4000. *http://www.donjohnston.com/catalog/cow4000d.htm* .

Demasco, Patrick W., and Kathleen F. McCoy. 1992. Generating text from compressed input: An intelligent interface for people with severe motor impairments. *Communications of the ACM* 35.

Embley, David W., M.T. Lan, Leinbaugh D.W., and George Nagy. 1978. A procedure for predicting program editor performance from the user’s point of verw. In *International Journal Man-Machine Studies 10: 639–650*.

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

Even-Zohar, Yair, and Dan Roth. 2000b. A classification approach to word prediction. In *The 1st Conference of the North American Chapter of the Association of Computational Linguistics (NAACL)*.

Even-Zohar, Yair, Dan Roth, and Dmitry Zelenko. 1999. Word prediction and clustering. In *Bar-Ilan Symposium on the Foundations of Artificial Intelligent*. Bar-Ilan, Israel.

Fazly, Afsaneh. 2002. *The use of syntax in word completion utilities*. Master’s thesis, Department of Computer Science, University of Toronto.

Fazly, Afsaneh, and Graeme Hirst. 2003. Testing the efficacy of part-of-speech information in word completion. In *Proceedings of the Workshop on Language Modeling for Text Entry Methods, 11th Conference of the European Chapter of the Association for Computational Linguistics*. Budapest, Hungary.

Foster, George, Philippe Langlais, and Guy Lapalme. 2002. User-friendly text prediction for translation. In *Proceedings of Intelligent User Interfaces*, 148–155. Philadelphia.

Garay-Vitoria, Nestor, and Julio Abascal. 2004. A comparison of prediction techniques to enhance the communication rate. In *Proceedings of the 8th ERCIM Workshop on User Interface for All*, 400–417.

Garay-Vitoria, Nestor, and Julio Gonz. 1997. Intelligent word-prediction to enhance text input rate (a syntactic analysis-based word-prediction aid for people with severe motor and speech disability). In *IUI '97: Proceedings of the 2nd international conference on Intelligent user interfaces*, 241–244. ACM Press.

Goldberg, Amie, Michael Russell, and Abigail Cook. 2003. The effect of computers on student writing: A meta-analysis of studies from 1992 to 2002. In *The Journal of Technology, Learning, and Assessment*.

Goodman, Joshua, Gina Venolia, Keith Steury, and Chauncey Parker. 2002. Language modelling for soft keyboards. In *The Eighteenth National Conference on Artificial Intelligence (AAAI-02)*, 419–424.

Handley-More, Dottie. 2001. Use of word precessors to support written communication: an annotated bibliography. *Physical and Occupational Therapy in Pediatrics* 21:5–17.

Harbusch, Karin, Saša Hasan, Hajo Hoffmann, Michael Kühn, and Bernhard Schüler. 2003. Domain-specific disambiguation for typing with ambiguous keyboards. In *Proceedings of the Workshop on Language Modeling for Text Entry Methods, 11th Conference of the European Chapter of the Association for Computational Linguistics*. Budapest, Hungary.

Higginbotham, D. Jeffery. 1992. Evaluation of keystroke savings across five assistive communication technologies. In *AAC Augmentative and Alternative Communication, 8: 258–272*.

Hyatt, Gwen, and Ann Black. 2005. Using technology to improve reading and writing outcomes for learning disabled students: a sequenced approach. In *Project Report of Special Education Technology Center. http://www.cwu.edu/ setc/ldtech/SequencedApproachReport.PDF*.

Iyer, R., and M. Ostendorf. 1996. Modeling long distance dependence in language: Topic mixtures vs. dynamic cache models. In *Proceedings of International Conference on Spoken Language Processing (ICSLP96)*, 236–239. Philadelphia, PA.

Jarmasz, Mario, and Stan Szpakowicz. 2001. Roget’s thesaurus: a lexical resource to treasure. In *Proceedings of the NAACL WordNet and Other Lexical Resources workshop*, 186–188.

Jelinek, Frederick, and Robert L. Mercer. 1980. Interpolated estimation of markov source parameters from sparse data. In *Proceedings of the Workshop on Pattern Recognition in Practice*.

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

Jiang, J.J., and D.W. Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of International Conference on Research on Computational Linguistics*.

Johansen, Anders Sewerin, Kenji Itoh, Satoru Mashino, John Paulin Hansen, and Dan Witzner Hansen. 2003. Language technology in a predictive, restricted on-screen keyboard with dynamic layout for severely disabled people. In *Proceedings of the Workshop on Language Modeling for Text Entry Methods, 11th Conference of the European Chapter of the Association for Computational Linguistics*. Budapest, Hungary.

Klarlund, Nils, and Michael Riley. 2003. Word *n*-grams for cluster keyboards. In *Proceedings of the Workshop on Language Modeling for Text Entry Methods, 11th Conference of the European Chapter of the Association for Computational Linguistics*. Budapest, Hungary.

Klund, Jamie. 1995. If word prediction can help, which program do you choose? *http://trace.wisc.edu/docs/wordprediction2001/index.htm* .

Koester, Heidi Horstmann, and Simon P. Levine. 1994. Modeling the speed of text entry with a word prediction interface. In *IEEE Transactions on Rehabilitation Engineering*, volume 2, 177–187.

Kondrak, Grzegorz. 2001. Indentifying cognates by phonetic and semantic similarity. In *Second Meeting of the North American Chapter of the Association for Computational Linguistics*.

Kozima, Hideki, and Akira Ito. 2004. A scene-based model of word prediction. In *citeseer.ist.psu.edu/97151.html*.

Kuhn, Roland, and Renato de Mori. 1990. A cache-based natural language model for speech recognition. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 570–583.

Laine, Colin J. 1998. Using word-prediction technology to improve the writing of low-functioning children and adults. In *The Australian Computers in Education Conference (ACEC)*.

Laine, Colin J. 2000a. Finding your way to effective technology: when can computers help? *International Special Education Congress* .

Laine, Colin J. 2000b. Using technology toolkit to cue written expression: which features help? what are the effects? In *Center On Disabilities Technology And Persons With Disabilities Conference*.

Laine, Colin J., and Tony Bristow. 1999. Using manual word-prediction technology to cue student's writing: does it really help? In *International Conference on Technology and Persons with Disabilities*.

Laine, Colin J., and R. Follansbee. 1994. Using word-prediction technology to improve the writing of low-functioning hearing-impaired students. In *Child Language Teaching and Therapy*.

Leacock, C., and Martin Chodorow. 1998. Combining local context and WordNet similarity for word sense identification. In *C. Fellbaum (ed.): WordNet: An Electronic Lexical Database, Chapter 11, pp. 265–283*.

Lesher, Gregory W., and Gerard J. Rinkus. 2005. Domain-specific word prediction for augmentative communication. *http://www.enkidu.net/downloads/papers/LeRi01.pdf* .

Lesk, Michael. 1986. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from a ice cream cone. In *Proceedings of the Fifth International Conference on Systems Documentation*, 24–26.

Li, Jianhua, and Graeme Hirst. 2005. Semantic knowledge in a word completion task. In *Proceedings, 7th International ACM SIGACCESS Conference on Computers and Accessibility*.

Magnuson, Tina, and Sheri Hunnicutt. 2002. Measuring the effectiveness of word prediction: The advantage of long-term use. In *TMH-QPSR, KTH, 43:57– 67. http://www.speech.kth.se/qpsr/tmh/2002/02-43-057-067.pdf* .

Manning, Christopher, and Hinrich Schütze. 2001. *Foundations of statistical natural language processing*. Cambridge, Massachusetts, London, England: The MIT Press.

Matiasek, Hohannes, and Baroni Marco. 2003. Exploiting long distance collocational relations in predictive typing. In *Proceedings of the Workshop on Language Modeling for Text Entry Methods, 11th Conference of the European Chapter of the Association for Computational Linguistics*. Budapest, Hungary.

Matiasek, Johannes, Marco Baroni, and Harald Trost. 2002. Fasty: A multi-lingual approach to text prediction. In *Proceedings of the 8th International Conference on Computers Helping People with Special Needs*.

McCoy, Kathleen, and Patrick Demasco. 1995. Some application of natural lanuage processing to the field of augmentative and alternative communication. In *Proceedings of the IJCAI-95 Workshop on Developing AI Application for People with Disabilities*.

McCoy, Kathleen, Christopher Pennington, and Arlene Luberoff Badman. 1998. Companion:

From research prototype to practical integration. In *Natural Language Engineering 4(1): 73–95*.

Miller, George A., and Walter Charles. 1991. Contextual correlates of semantic similarity. *Language and Cognitive Pocesses* 6.

Patwardhan, Siddharth, Satanjeev Banerjee, and Ted Pedersen. 2003. Using measures of semantic relatedness for word sense disambiguation. In *Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics*. Mexico City.

Purandare, Amruta, and Ted Pedersen. 2004. Word sense discrimination by clustering contexts in vector and similarity spaces. In *Proceedings of the Conference on Computational Natural Language Learning(CoNLL)*.

Renaud, Alfred. 2002. *Diagnostic evaluation measures for improving performance of word prediction systems*. Master's thesis, School of Computer Science, University of Waterloo.

Resnik, P. 1995. Using information content to evaluate semantic similarity. In *Proceedings of the 14th International Joint Conference on Artificial Intelligent (IJCAI-95)*, 448–453.

Rosenfeld, Ronald. 1996. A maximum entropy approach to adaptive statistical language modeling. In *Computer, Speech and Language*, 10:187–228.

Rubenstein, Herbert, and John B. Goodenough. 1965. Contextual correlates of synonymy. *Communications of the ACM* 8:627–633.

Seymore, Kristie, Stan Chen, and Ronald Rosenfeld. 1998. Nonlinear interpolation of topic models for language model adaptation. In *Proceedings of International Conference on Spoken Language Processing (ICSLP98)*.

Shieber, Stuart M., and Ellie Baker. 2003. Abbreviated text input. In *Proceedings of the International Conference on Intelligent User Interfaces*, 293–296.

Sparck Jones, Karen, and Julia R. Galliers. 1996. *Evaluating natural language processing systems*. Springer.

Trnka, Keith, Debra Yarrington, Kathleen McCoy, and Christopher Pennington. 2006. Topic modeling in fringe word prediction for AAC. In *IUI '06: Proceedings of the 11th international conference on intelligent user interfaces*, 276–278.

Tsandilas, Thephanis, and m. c. Schraefel. 2005. An empirical assessment of adaptation techniques. In *Conference on Human Factors in Computing Systems, CHI '05 extended abstracts on Human factors in computing systems*.

Venolia, Gina, Keith Steury, and Chauncey Parker. 2002. Language modeling for soft keyboards. In *Proceedings of the International Conference on Intelligent User Interfaces*. San Francisco.

Ward, David J., Alan F. Blackwell, and David J.C. MacKay. 2005. Dasher – a data entry interface using continuous gestures and language models. *http://www.inference.phy.cam.ac.uk/dasher/* .

Wester, Malin. 2003. User evaluation of a word prediction system. In *Master's thesis*. Uppsala University.

Wigdor, Daniel, and Ravin Balakrishnan. 2004. A comparison of consecutive and concurrent input text entry techniques for mobile phones. In *Proceedings of the SIGCHI conference on Human factors in computing systems*.

Wood, Matthew E.J., and Eric Lewis. 1996. Windmill – the use of parsing algorithm to produce predictions for disabled persons. In *Proceedings of the 1996 Autumn Conference on Speech and Hearing*, 118: 315–322. Institute of Acoustics.

Word-Prediction. 2005. Increasing literacy levels by the use of linguistic prediction. *http://www2.edc.org/NCIP/library/wp/Newell.htm* .

WordQ. 2004. WordQ writing aid software. *http://www.wordq.com* .

Zhang, Y., D. Brooks, T. Frields, and M. Redelfs. 1995. Quality of writing by elementary students with learning disabilities. In *Journal of Research on Computing in Education*, 27:483–499.