Breaking the Language Barrier:
A Game-Changing Approach

Version 0.02

Yao Ziyuan
yaoziyuan@gmail.com
http://sites.google.com/site/yaoziyuan/

Apr 1, 2010

Table of Contents

Overview

This material introduces pioneering ideas, many of them rarely noticed, that will redefine the way people break the language barrier, and organizes them into a big picture as illustrated by the Table of Contents.

The grand problem of how to break the language barrier can be divided to two subproblems: one that involves the user in serious learning of a foreign language and one that doesn't. Chapters 1 and 2 address these two subproblems respectively.

When we also take into account human resources available online that could help with summarized or full translation, there is the question of how to make good use of them, which is discussed in Chapter 3.

Ideas whose titles have an exclamation mark (!) are stirring game-changing technologies which are the driving forces behind this grand initiative.

Chapter 1: Breaking the Language Barrier with Language Learning

Sometimes a person wants to internalize a foreign language in order to understand and generate information in that language, especially in the case of English, which is the de facto lingua franca in this globalized era.

Section 1.1 “Foreign Language Acquisition” discusses a novel approach to learning a foreign language (exemplified by English).

A person with some foreign language knowledge may still need assistance to better read and write in that language. Therefore, Sections 1.2 “Foreign Language Writing Aids” and 1.3 “Foreign Language Reading Aids” discuss how novel tools can assist a non-native user in writing and reading.

1.1. Foreign Language Acquisition

A language can be divided into two parts: the easy part is its grammar and a few function words, which account for a very small and fixed portion of the language's entire body of knowledge; the hard part is its vast vocabulary, which is constantly growing and changing and can't be exhausted even by a native speaker.

Therefore, the problem of language acquisition is largely the problem of vocabulary acquisition, and a language acquisition solution's overall performance is largely determined by its vocabulary acquisition performance.

The problem of vocabulary acquisition can be divided into two subproblems: “when” – when is potentially the best time to teach the user a word, and “how” – when such a teaching opportunity comes, what is the best way for the user to memorize the word and bond its spelling, pronunciation and meaning all together?

Section 1.1.1 addresses the “when” problem with a method called automatic code-switching, which smartly administers the user's vocabulary acquisition experience and applies to grammar acquisition as well.

Section 1.1.2 addresses the “how” problem with various mnemonic devices starring “Phonetics-Enhanced English”, all of them fitting neatly with the automatic code-switching framework.

1.1.1. Automatic Code-Switching! (ACS)

Automatic code-switching is a computer-based foreign language acquisition strategy that sprays relevant foreign language elements sporadically in the user's native language communication.

A Quick Introduction

The computer automatically selects a few words in a user's native language communication (such as a Web page being viewed), and supplements or even replaces them with their foreign language counterparts, thus naturally building up his vocabulary. For example, if a sentence

他是一个好学生。

(Chinese for “He is a good student.”) appears in a Chinese person's Web browser, the computer can insert “student” after “学生” (optionally with additional information such as student's pronunciation):

他是一个好学生 (student)。

After several times of such teaching, the computer can directly replace future occurrences of “学生” with “student”:

他是一个好 student。

Ambiguous words such as the “看” (Chinese for “see”, “look”, “watch”, “read”, etc.) in

他在电视前看书。

(Chinese for “He is reading a book before the TV.”) can also be automatically handled by listing all context-possible translations:

他在电视前看 (阅读: read; 观看: watch) 书。

Practice is also possible:

他在电视前 [read? watch?] 书。

Because the computer would only teach and/or practice foreign language elements at a small number of positions in the native language article the user is viewing, the user wouldn't find it too intrusive. Automatic code-switching can also teach grammatical knowledge in similar ways.

A Linguistic Concern and Its Solution

A criticism to code-switching is even if a foreign language element is synonymous with the native language element it replaces, it may not fit into the native language sentence syntactically. For example, what if the native language element is an intransitive verb but the foreign language element is a transitive verb? The computer should also try to transform the native verb's argument structure (complements, prepositional phrases and other constituents that belong to the verb phrase) to that of the foreign verb. If it fails to do so (due to unresolvable syntactic ambiguity in analyzing the native verb's argument structure), the computer can add remarks in parentheses after the foreign language element to explain its syntax usage.

An Analysis from an NLP/MT Researcher's Perspective

Apparently the implementation of an ACS system shares fundamental techniques with machine translation and natural language processing in general. Word sense disambiguation (WSD) and syntax analysis research from these fields can be immediately reused in the emerging field of ACS. For example, we can order the two alternative senses “read” and “watch” in the above example according to their probability of appearing in that particular context calculated by a statistical WSD algorithm.

Yet ACS enjoys a lot more freedom than MT. It is actually “partial”, or even “sparse”, machine translation, with additional technical advantages. First, the user's cognitive threshold mandates that only a small percent of the whole article's words be taught, and the machine can choose which words to teach, and therefore the machine can prefer those unambiguous words and those words it has a high confidence in disambiguating. Second, even poorly disambiguated words can be taught/practiced by listing the two or three most likely senses. More senses can be hidden in a user interface (UI) element that means “more” (e.g. “…”, “►”, and “»”) and can be shown when the user clicks or moves the mouse on that element.

他在电视前 [read? watch? »] 书。

Third, even if we don't have good context-based WSD capabilities to order senses according to their context feasibility, we can still simply order them by their frequency in a large corpus. We expect the top two or three senses combined to account for most cases, and the user will only occasionally have to check less used senses hidden in “more”.

For grammatical knowledge of the foreign language, the machine can find a portion of the text (a word, a phrase, a clause or a sentence) which has an unambiguous or disambiguatable grammatical feature, and teach/practice that feature. Theoretically, ambiguous grammatical features can also be handled by list all possible translations just like ambiguous words, but apparently should be less used.

It should also be noted that ambiguity is not always an enemy to ACS as it is to machine translation, because lexical/syntactic ambiguity is exactly a kind of natural language phenomenon that we want the student aware of. So pointing out ambiguity is sometimes beneficial to the student.

ACS in a Multi-Party Environment

If an ACS system transforms not only the user's incoming communication (e.g. a Web page loaded or an email/IRC/instant-messenger (IM) message received) but also outgoing communication (e.g. a post to a forum or blog, or an email/IRC/IM message sent), all his recipients will be engaged in language learning, even if they themselves do not install an ACS system on their client side. Put another way, if only one active participant in an online community (e.g. an IRC chatroom or a forum) ACS-izes his messages, all other members will be learning the foreign language. It's like someone smoking in a lobby – no one else will survive the smoke.

Such a situation also fosters language learners' "productive knowledge" in addition to "receptive knowledge" ("receptive" means a learner can recognize the meaning of a word when he sees or hears it, while "productive" means he can independently write or say a word when he wants to express the meaning). For example, suppose two Chinese Hong and Ming are chatting with each other, and Hong says:

他是一个好学生。

(Chinese for "He is a good student."), but Hong's client-side ACS system transforms this outgoing message, replacing the Chinese word "学生" with its English counterpart "student":

他是一个好 student。

Now both Hong and Ming see this transformed message, and suppose Ming wants to say:

他不是一个好学生。

(Chinese for "He is not a good student."), but he is influenced by the English word "student" in Hong's message, and subconsciously follows suit, typing "student" instead of "学生" in his reply:

他不是一个好 student。

Thus, Ming is engaged in not only "recognizing" this English word but also "producing" it, although not based on independent recall from his own memory.

Independent exercise of productive knowledge can be induced if Hong's ACS system transforms her original message in another way:

他是一个好 s____ (学生)。

which looks as if Hong wanted to express the concept in English but could only recall the first letter, so she expressed the concept in Chinese in the parenthesis, leaving the incomplete English word alone. If Ming is also going to refer to this concept, Hong's "failed attempt" may inspire him to complete the challenge.

Historical Notes

1960s: Code-switching as a method for language education was first proposed by American anthropologist and linguist Robbins Burling, and has largely remained a handicraft – foreign language elements are added by human editors rather than computers. Burling dubbed it "diglot reader/diglot weave" and was inspired by a "Learning Chinese" book series published by Yale University Press, where new Chinese characters gradually replaced Romanized Chinese in a text.

1960s – Present: In academic literature, manual code-switching is almost solely maintained by Brigham Young University researchers (search Google Scholar for “diglot reader”, “diglot weave” and “diglot method”).

1990s – Present: There are educational materials using manual code-switching but they have never gone mainstream: PowerGlide (www.power-glide.com) and “Three Little Pigs and Stepwise English” (三只小猪进阶英语) by Professors Ji Yuhua (纪玉华) and Xu Qichao (许其潮).

2004 – Present: I independently came up with the code-switching idea and researched it as an automatic approach from the beginning. Research notes have been posted to the Usenet newsgroup list.linguist since Oct 2004. Major aspects of this research are discussed in this section. I also conceived a name for such a system: ATLAS – Active Target Language Acquisition System.

2006: WebVocab (http://webvocab.sourceforge.net/) is another attempt to automatic code-switching, but its development discontinued years ago and it only disambiguates words by function-word clues (e.g. a word after “I” must be a verb/adverb rather than a noun/adjective, so “can” after “I” must be in the auxiliary verb sense rather than a container), and otherwise it will not teach or practice ambiguous words at all.

1.1.2. Mnemonics

The automatic code-switching (ACS) framework discussed in Section 1.1.1 already implies an approach to word memorization: by repetition (a new word is taught and practiced several times in context before it is considered learned). Research into more sophisticated mnemonics has unveiled methods that can serve as powerful force multipliers for the vanilla ACS approach.

Mnemonics discussed below can be categorized into several strategies:

Divide-and-Conquer: Sections 1.1.2.3 “Progressive Word Acquisition” and 1.1.2.4 “Subword Familiarization” are approaches that split a long word into more digestible or more familiar parts and eventually conquer the whole word.

Compress-and-Conquer: Memorizing a word in terms of syllables takes far less effort than in terms of letters, and therefore pronunciation as a more compressed form than spelling is a key mnemonic. Section 1.1.2.1 “Phonetics-Enhanced English” is an approach that exposes a word's pronunciation as early as its spelling to the learner, enforcing correct, confident and firm acquisition of pronunciation, which in turn effectively facilitates memorization and recall of spelling.

Amplify-and-Conquer: Certain parts of a long word can be so obscure that they are often ignored even by native speakers, such as a word's choice between “-ance” and “-ence”. Section 1.1.2.2 “Orthography-Enhanced English” discusses an approach that deliberately “amplifies” such “weak signals”, so that the learner gets a stronger impression.

Ad-Hoc: Mnemonics in general is about converting the information to remember to more familiar or easier-to-remember information. Depending on an individual's cultural background and personal experience, familiar information varies. Therefore everyone may have his own ad-hoc mnemonic for a particular word, which is discussed in Section 1.1.2.5.

Among them, the most outstanding approach is “Phonetics-Enhanced English”, although other approaches can also exhibit exceptional performance.

1.1.2.1. Phonetics-Enhanced English! (PEE)

Phonetics-Enhanced English slightly decorates or modifies an English word's visual form (usually by adding diacritical marks) to better reflect its pronunciation, while retaining its original spelling. This is used in early stages of word acquisition by a non-native learner (preferably administered by automatic code-switching), in order to enforce correct, confident and firm memorization of pronunciation as early as possible, which in turn also facilitates effective memorization of spelling.

A Quick Introduction
A full-fledged PEE sentence may look like this:

A qu̦ic̦k brőw̵n fox jumps ọ̄ve͠r ț h̵è lạ̄zy̆ dog.

The above example shows pronunciation in the most verbose mode: “u”, “c”, “o”, “t”, “e”, “a” and “y” are assigned diacritics to differentiate from their default sound values; “w” and “h” have a short bar which mean they're silent; “er” bears a “double-letter diacritic” to indicate a schwa; multi-syllable words such as “over” and “lazy” have a dot to indicate stress. Such a mode is intended for a non-native beginner of English, who is unaware of digraphs like “ow”, “er” and “th”.

On the other hand, more advanced learners can use a liter version:

A qu̦ick brőwn fox jumps ọ̄ver țhè lạ̄zy dog.

Furthermore, words and word parts (e.g. -tion) that a learner is already familiar with also don't need diacritics.

PEE is ideally used in conjunction with automatic code-switching, a foreign language learning framework that administers foreign language elements in a learner's native language communication (see Section 1.1.1 for details).

Advantages
One advantage of PEE over other phonetic transcription schemes such as the IPA (International Phonetic Alphabet) and the various respelling systems used in major American dictionary brands is that phonetic information is “integrated” with spelling so as to provide “immediate phonetic awareness”, in contrast to “separate transcriptions” that require the learner to “look somewhere else” in the process of reading something, solely for the seemingly no-so-necessary objective of pronunciation acquisition – after all, the learner is more interested in the meaning, instead of pronunciation, of a new word. Learning pronunciation is deemed “optional” or even “a waste of time” for most non-native learners who “only need to deal with the foreign language textually”.

PEE not only enforces correct and repeated instruction of pronunciation, but also has the unexpected effect of facilitating acquisition of spelling. For example, suppose a non-native learner encounters the word “thesaurus” for the first time in reading. He can easily look up its meaning with a “point-to-translate” dictionary program (i.e. just move the mouse onto that word and its translation will be shown, as seen in Google Toolbar's “word translation” feature), but won't bother looking at its pronunciation because, as discussed above, pronunciation is unnecessary. For future occurrences of

"thesaurus", the learner may be able to recall its meaning without resorting to the dictionary again. He may seem to have "fully mastered" the word. But not really – if you say the word's meaning in his native language (e.g. Chinese) to him and ask him to independently write the corresponding English word down, he would almost certainly stumble upon it. The fact is he almost never tries to memorize the full spelling of "thesaurus" because there is no need – he actually just memorizes a "shorthand", e.g. "thes...s", and this abbreviated form, in conjunction with the context, is sufficient to uniquely represent the English word ("thesaurus", although he doesn't know this full form) in his mind, and upon seeing this abbreviation, he can retrieve the meaning associated with it. On the other hand, if he does try to memorize the word's full spelling in reading but still doesn't bother to see its pronunciation, there are two options, neither of them effective. The first option is to come up with a guessed pronunciation and use that pronunciation as a key to recall the spelling. The problem is, without systematic training in English phonology ("rational knowledge") or extensive word pronunciation samples like a native speaker has ("empirical knowledge"), the guessed pronunciation is often wrong, and wrong pronunciations may require a great effort to get "unlearned" in the future. Afraid of this, the learner doesn't dare to "commit" the guessed pronunciation to his memory firmly, resulting only a shallow memory footprint. Because it's shallow, it could soon be forgotten entirely, taking him back to square one. The chilling effect is very likely to prevent him from guessing a full pronunciation at all, resulting in an abbreviated form of the word like "thes...s" discussed above. The second way is stupider and less common: to memorize the spelling letter by letter, without any pronunciation: T-H-E-S-A-U-R-U-S. This has the same level of cognitive complexity as that of remembering a telephone number or ICQ number, which can be prohibitively hard compared to remembering the three syllables in the phonetic form.

A Disadvantage and How to Reduce It

The only noticeable disadvantage of PEE is it may look "cumbersome" compared to no diacritics at all. As discussed earlier in "A Quick Introduction", the cumbersomeness can be reduced by switching to liter versions of PEE as the learner advances his study, and by dropping diacritics from words and word parts that the learner is already familiar with.

Technical Analysis

There are several technical approaches to "adding something above normal text". You can design a special font that draws letters with diacritics, or Web browser extensions, plugins and server-side scripts that dynamically generate graphics from special codes (e.g. MathML), or HTML "inline tables" as used in an implementation of "Ruby text" (http://en.wikipedia.org/wiki/Ruby_character, http://web.nickshanks.com/stylesheets/ruby.css), or systems that use two Unicode features – "pre-composed characters" (letters that come with diacritics right out of the box) and "combining codepoints" (special characters that don't stand alone but add diacritics to other characters).

Attached to the Appendix of this material is a sample PEE scheme that uses both Unicode pre-composed characters and combining codepoints.

In the making of this sample scheme, I consulted these sources:

- *English spelling* (http://en.wikipedia.org/wiki/English_spelling#Spelling_patterns): spelling-to-sound and sound-to-spelling patterns in English.
- *Combing character* (http://en.wikipedia.org/wiki/Combining_character): tables of combining characters in Unicode.
- *Pronunciation respelling for English* (http://en.wikipedia.org/wiki/Pronunciation_respelling_for_English): Comparison of respelling

schemes in major dictionaries.

It should be noted that a font optimized for Unicode (e.g. the free and open source DejaVu Sans) is required to render Unicode characters used in this scheme properly. Therefore a Web browser extension that converts normal English words to PEE should also enforce Unicode-optimized fonts on PEE text.

Historical Notes
The general idea of representing a letter's various sound values by additional marks is probably as old as diacritics.

American dictionaries before the 20th century showed diacritical marks directly above headwords to indicate pronunciation for native readers (though not necessarily verbosely). They have been replaced by separate transcription schemes, such as the IPA and respelling systems.

Adding diacritics to English for non-native learners is an obscure method that has never gone mainstream. In 2000s China, Professor Sun Wenkang (孙文抗) reviewed previous work and devised a scheme called "EDS", and published a now out-of-print book "Categorized Basic English Vocabulary with EDS" (EDS 注音基础英语分类词汇手册).

I independently came up with this idea in Mar 2009 and created a scheme called "Phonetics-Enhanced English" (PEE), based on Unicode. The scheme is attached to the Appendix of this material.

1.1.2.2. Orthography-Enhanced English (OEE)

PEE in Section 1.1.2.1 essentially encodes a word's pronunciation into its spelling. This spawns a symmetric question: can we encode a word's spelling into its pronunciation as well? For example, "reference" and "insurance" have suffixes that sound the same but spell differently (-ence and -ance), and can we differentiate these suffixes' pronunciations to reflect their spelling difference? Can we give -ance a rising tone and -ence a falling tone? This sounds Chinese and would create new dialects for English that lead to chaos in conversations.

However, if we think outside the box, if we no longer try to "put information into pronunciation", we may be able to explore other avenues. What about putting this information into a word's visual form? What about lowering the "a" in -ance a little so that it makes a different impression on the learner? So we have

insur$_{a}$nce

in contrast to

reference

Makes a difference, doesn't it? If the learner develops visual memory that "insurance" has a lowered character in its suffix, then he can infer that this suffix is -ance because -ance has a lowered "a" while -ence doesn't have anything lowered.

We call this "Orthography-Enhanced English" (OEE).

Technical Analysis
Like PEE, OEE makes slight modifications to a word's visual form to add some extra information. Therefore they can share the same techniques for rendering to such effects. In the above example, most

document formats that support rich formatting should allow us to raise/lower a character from its baseline. Particularly, in HTML, we can use the tag and its "vertical-align" style property:

```
<p>insur<span style="vertical-align: -15%">a</span>nce</p>
```

which will lower the "a" in "insurance" by 15%, making the word look like

insur$_{a}$nce

Of course, we can also encapsulate the style property into a CSS class so that the above HTML code can shrink to something like

```
<p>insur<span class="lowered">a</span>nce</p>
```

Historical Notes

Nothing really much to see here. I haven't found previous work for this idea. I came up with it in Aug 2009.

1.1.2.3. Progressive Word Acquisition (PWA)

In automatic code-switching (see Section 1.1.1), long words are optionally split into small segments (usually two syllables long) and taught progressively, and even practiced progressively. For example, when

科罗拉多州

(Chinese for "Colorado") first appears in a Chinese person's Web browser, the computer inserts Colo' after it (optionally with Colo's pronunciation):

科罗拉多州 (Colo')

When 科罗拉多州 appears for the second time, the computer may decide to test the user's memory about Colo' so it replaces 科罗拉多州 with

Colo' (US state)

Note that a hint such as "US state" is necessary in order to differentiate this Colo' from other words beginning with Colo.

For the third occurrence of 科罗拉多州, the computer teaches the full form, Colorado, by inserting it after the Chinese occurrence:

科罗拉多州 (Colorado)

At the fourth time, the computer may totally replace 科罗拉多州 with

Colorado

Not only the foreign language element (Colorado) can emerge gradually, the original native language element (科罗拉多州) can also gradually fade out, either visually or semantically (e.g. 科罗拉多州 → 美国某州 → 地名 → ∅, which means Colorado → US state → place name → ∅). This prevents the learner from suddenly losing the Chinese clue, while also engages him in active recalls of the occurrence's complete meaning (科罗拉多州) with gradually reduced clues.

1.1.2.4. Subword Familiarization (SWF)

Again in automatic code-switching (see Section 1.1.1), word roots (e.g. pro-, scrib-) and meaningless word fragments (e.g. -ot) are optionally treated as two special kinds of standalone words and taught and

practiced just like normal words in the user's incoming native language information. Meaningful word roots are considered synonyms for their real word counterparts (e.g. pro- is considered a synonym for “advance”, “improve” and “support”) while meaningless fragments are considered abbreviations or acronyms derived from real words (e.g. -ot is considered an acronym for “off topic”).

Getting the learner familiar with such subword units in advance can facilitate future acquisition of longer, real words that contain them.

Even if a subword fragment is not semantically related to a long word that contains it, it still serves as a cornerstone for memorizing the entire word's spelling.

1.1.2.5. Ad-Hoc Mnemonics

Mnemonics in general is about converting the information to remember to more familiar or easier-to-remember information. Depending on an individual's cultural background and personal experience, familiar information varies. Therefore everyone may have his own ad-hoc mnemonic for a particular word.

For example, as a Chinese, when I first encountered the word “sonata” in a multimedia encyclopedia as a teenager, I associated it with a traditional Chinese musical instrument Suona (唢呐) which was featured in an elementary school music class and bears a similar pronunciation to the “sona” part of “sonata”.

Therefore it is useful to let people from the same cultural background contribute their own mnemonics collaboratively online. Wiktionary might be a potential site for such collaboration. People from different cultural backgrounds can also exchange mnemonics based on common knowledge, like using English words as mnemonics for English words.

1.1.3. Sociopsychological Considerations

Beside technical aspects in foreign language acquisition discussed in Sections 1.1.1 and 1.1.2, there are non-technical problems that may affect a non-native learner's language and social experience. Sections 1.1.3.1 and 1.1.3.2 discuss two such problems and possible solutions.

1.1.3.1. A Politically Correct Name for English (ARCS)

As technology like automatic code-switching would make English a much cheaper commodity for non-native people to acquire, for the first time it will become possible for most people in the world to use decent English. But nationalist sentiments can be a negative factor for some people to adopt English.

While it is logically recognized by everybody that all natural languages are actually made of equally random syllables, emotionally people can still more or less feel unequal that one language is more international than others. A cause for this paradox is that languages are named by their nations of origin: English, French, Spanish, etc. Accordingly, we can use a “renaming” technique to better reflect a language's random nature rather than nationalist connotation. Actually, the word “language” itself already has a strong nationalist connotation, and I propose the term “code system” to eliminate that connotation. As for English, let's rename it as “A Random Code System”, or ARCS for short.

1.1.3.2. Intuitive Scores for Language Proficiency (FLPM)

How does a non-native speaker introduce his language level to a native speaker in an understandable manner? The native speaker may not likely know what a TOEFL or IELTS score really means.

The computer can test the non-native speaker's language proficiency and compare it with native speakers at different ages. Introductions like "My English level is like a 10-year-old American child" should be understood well by a native speaker.

Of course, such measurement can be further split to components like reading, writing, speaking and listening.

1.2. Foreign Language Writing Aids

A person with some foreign language knowledge may still need assistance to better write in that language. This section discusses how novel tools can assist a non-native user in writing.

1.2.1. Predictive vs. Corrective Writing Aids

In contrary to foreign language acquisition methods such as automatic code-switching which builds up the user's language knowledge preparatively, the user may need on-demand language support upon reading or writing something totally in a foreign language. This is especially true of writing, which requires “productive knowledge” that is often ignored in reading, such as a word's correct syntax and applicable context.

On-demand writing aids can be divided into two types:

Predictive writing aids predicts lexical, syntactic and topical information that might be useful in the upcoming writing, based on clues in previous context. Section 1.2.2 discusses two such tools.

Corrective writing aids retroactively examines what is just inputted for possible errors and suggestions. A spell checker is a typical example, which checks for misspellings in input. Corrective writing aids is a much researched area, as most natural language analysis techniques can be applied to examine sentences for invalid occurrences, and there are studies on non-native writing phenomena such as wrong collocations. Therefore this material does not expand this topic.

1.2.2. Input-Driven Syntax Aid! (IDSA)

As a non-native English user inputs a word, e.g. “search”, the word's sentence-making syntaxes are prompted by the computer, e.g.

v. search: n. searcher search... [n. search scope] [for n. search target]

which means the syntax for the verb “search” normally begins with a noun phrase, the searcher, which is followed by the verb's finite form, then by an optional noun phrase which is the search scope, and then by an optional prepositional phrase stating the search target.

With this information, the user can now write a syntactically valid sentence like

I'm searching the room for the cat.

1.2.3. Input-Driven Ontology Aid! (IDOA)

As a non-native English user inputs a word, e.g. “badminton”, things (objects) and relations that normally co-exist with the word in the same scenario or domain are prompted as a systematic ontology graph (semantic network) by the computer, which include objects like “racquet”, “shuttlecock” and “playing court”, relations like “serve” and “strike”, and even full-scripted essay templates like “template: a badminton game”.

The benefits of the ontology aid are twofold. First, the ontology helps the user verify that the "seed word", badminton, is a valid concept in his intended scenario (or context); second, the ontology pre-emptively exposes other valid words in this context to the user, preventing him from using a wrong word, e.g. bat (instead of racquet), from the very beginning.

1.3. Foreign Language Reading Aids

Unlike non-native writing, non-native reading doesn't require much help from sophisticated tools. A learner with basic English grammar and the most frequent 100-300 words can engage in serious reading with the help from a point-to-translate dictionary program (the program shows translations for whatever English word is under the learner's mouse).

It should be noted that in reading something the learner only cares about the meaning of an unfamiliar word, not further information such as irregular verb forms. Such further information is taught in automatic code-switching or timely provided by writing aids, but can also be introduced using the approach below.

A reading aid can insert educational information about a word or sentence into the text being read, just like automatic code-switching, with the only difference that the main text is in the foreign language rather than the native language. This enables the computer to teach additional knowledge such as idioms and grammatical usages that are beyond word-for-word translation. Word-specific syntaxes as discussed in Section 1.2.2 “Input-Driven Syntax Aid” and domain-specific vocabularies as discussed in Section 1.2.3 “Input-Driven Ontology Aid” are also good feeds.

Chapter 2: Breaking the Language Barrier with Little Learning

Sometimes a person doesn't intend to acquire the vast vocabularies of the world's many languages, as English alone is already massive. He more likely would like to harness the computer's memory capacity to interpret and generate words in those other foreign languages. On the other hand, it seems advantageous if we let the human take care of syntax analysis and synthesis. Sections 2.1 and 2.2 discuss how the human and the machine can work together to understand and generate information in a foreign language.

2.1. Foreign Language Understanding

Section 2.1.1 introduces a new approach to decoding information in a foreign language where the computer takes care of content word translation and the human takes care of syntax understanding.

2.1.1. Machine Translation with Minimum Human Learning! (MT/MHL)

Before artificial intelligence reaches its fullest potential, machine translation always faces unresolvable ambiguities. The good news is, statistical MT such as Google Translate disambiguates content words quite well in most cases, and syntactic ambiguity can largely be "transferred" to the target language, without being resolved, if both the source and the target language have common syntactic features. For example, both English and French support prepositional phrases, so

I passed the test with his help.

can be translated to French without determining whether "with his help" modifies "passed" or "the test". The bad news is, syntactic disambiguation usually can't be bypassed between a language pair like English to Chinese, as in Chinese you must determine which constituent is modified by "with the help", and put "with his help" before that constituent ("passed" or "the test"). Resolution of syntactic ambiguity requires capabilities ranging from shallow rules (e.g. "with help" should modify an action rather than an entity, and if there are more than one action – both "pass" and "test" can be considered actions – it should modify the verb – "pass") to the most sophisticated reasoning based on context or even information external to the text (e.g. in

Do you see the cat near the tree and the man?

what is the prepositional object of "near"? "The tree" or "the tree and the man"?)

In light of automatic code-switching, an emerging technology that promises to make foreign language learning efficient and effortless (see Section 1.1.1), we can actually prepare a human with essential syntactic knowledge (including key prepositions) of a foreign language (or language family), so that machine translation can simply take care of content word translation (in case a word's default translation doesn't make sense, the user can move the mouse to that translation to see alternative translations) and leave syntactic puzzles "as is" to the human. For example, when translating

I passed the test with his help.

to Chinese, instead of trying to determine which word is modified by "with his help", the MT system simply retains the original word order and preposition ("with"), and just translates the content words:

我 通过了 测试 with 他的 帮助。

which literally means "I PASSED THE-TEST with HIS HELP."

Further, we can devise and employ an “International Grammatical Alphabet” analogous to the International Phonetic Alphabet, to have a unified way to mark in a sentence:

- grammatical constituents such as the subject, the main verb, the object, the manner, the location and the time;
- dependency relationships between two constituents, such as whether a constituent modifies another constituent on the left/right of it, and whether this is an immediate left/right;
- part-of-speeches of function words when possible, such as preposition, postposition and circumposition;
- potentially ambiguous grammatical constructs specific to a language, such as English's “that”, “be”, “have”, “-ing”, “-ed” and “-s”;
- other grammatical relationships or features.

So the user can just learn a single set of grammatical symbols and transparently decode grammatical features in a foreign language sentence with these symbols.

2.2. Foreign Language Generation

Section 2.2.1 introduces an approach to encoding information in a foreign language where the human takes care of syntax well-formedness and the computer takes care of lexical disambiguation under the human's supervision.

2.2.1. Formal Language Writing and Machine Translation! (FLW)

A person not knowing a target language can generate information in that language by composing in a formal language based on his native vocabulary and having the composition machine-translated. Tools such as the input-driven syntax aid (see Section 1.2.2) and input-driven ontology aid (see Section 1.2.3) can be borrowed to assist the person in formal language writing.

If the user's native language is English, a formal language sentence based on English vocabulary may look like:

```
A quick brown fox.jump(over: the lazy dog);
```

which resembles an object-oriented programming language's function call, where “over” is an argument name for the function “jump” that is a member of the object “fox”, and “the lazy dog” is the value for that argument.

If a word in such a formal-syntax sentence is ambiguous, automatic word sense disambiguation (WSD) methods can calculate the most likely sense and immediately inform the user of this calculated sense by displaying a synonym in place of the original word according to this sense. The user can manually reselect a sense if the machine-calculated sense is wrong. All multi-sense content words are initially marked as “unconfirmed” (e.g. using underlines), which means their machine-calculated senses are subject to automatic change if later-inputted text suggests a better interpretation. An unconfirmed word becomes confirmed when the user corrects the machine-calculated sense of that word or some word after it (both cases imply the user has checked the word), or when the user hits a special key to make all currently unconfirmed words confirmed. This process is like how people input and confirm a Chinese string with a Chinese input method.

After all lexical ambiguity is resolved either automatically or manually, the computer can proceed to machine-translating the formal language composition to any target natural language.

Historical Notes
There are quite a few attempts at this approach. The most notable one is the UNL (Universal Networking Language) at http://www.undl.org.

I independently came up with this idea in 2003, by the end of high school.

Chapter 3: Breaking the Language Barrier with Collective Intelligence

When we take into account human resources available online that could help with summarized or full translation, there is the question of how to make good use of them, which is discussed in Section 3.1.

3.1. Collaborative Summarization and Translation

We first discuss how to uniquely identify a piece of source language information and let interested human translators and readers rendezvous at a single online location for collaboration, and then discuss how the translation and consumption process would happen.

Uniquely Addressing a Source Language Message
In this Web 2.0 era, the same piece of information (hereinafter "message") can spread and be copied to multiple online locations. Therefore there is the question of how to track these copies.

A message can be identified by a unique keyword combination (when the message has slightly different copies, like an academic paper available in differently formatted PDF, PS, DOC and HTML documents), or by a hash value of its entire content (when we know the message's copies are exactly the same).

With such identification information, a content indexing service like Google can allow readers of different copies of the same message to rendezvous at a unique location, where collaborative translation can take place.

Collaborative Summarization, Translation and Exploration
After interested readers gather at the rendezvous point, some of them may be able to translate the original message to a language more familiar to this group. If the message is important, it may get fully translated. Otherwise, the group may take an iterative approach. At first, one translator may contribute a one-line translation that summarizes the whole message, just like movie fans at imdb.com are willing to contribute a "tagline" for a movie (altough a tagline is not necessarily a summary). Subsequently, another translator may write a paragraph which is a more detailed summary, just like writing a short plot for a movie. Further, translators may translate aspects of the message that they feel interesting or that are requested by non-translators. Discussions and debates can happen, in which more information may be requested and provided, either by translating aspects in the message or looking elsewhere on the Web.

Appendix

A.1. The Phonetics-Enhanced English Scheme ver. 0.01

Warning

You need a Unicode-optimized font (e.g. the free and open source DejaVu Sans) to properly display the following content. From the end user's perspective, PEE is used in conjunction with automatic code-switching, where newly taught words are automatically transformed to PEE by a Web browser extension, and Unicode-optimized fonts are automatically enforced on PEE text.

Quick Examples

Full-fledged: A qu̦içk brőw̵n fox jumps ọ̄ve͠r țħè lạ̄zy̆ dog.
Lite: A qu̦ick brőwn fox jumps ọ̄ver țhè lạ̄zy dog.

Usage

The user is supposed to learn this PEE scheme "by example", i.e. they will know how new words sound by looking at diacritics used in known words. They don't need to systematically study the rules in the scheme, although gradual rule instruction in the context of reading (by an automatic code-switching system) is definitely a boost.

Depending on the user's level of English phonics knowledge, liter versions of this scheme can be used. Moreover, already acquired words and word parts (e.g. -tion) don't need diacritics.

Design Remarks

Version 0.01 is a working but not optimized scheme. Some diacritics used in this version could be replaced by ones that are visually clearer in some fonts, and diacritic assignment could be more scientific, logical and memorable.

Diacritics can be assigned in several ways: (1) each diacritic corresponds to a phoneme, regardless of the letter modified; (2) each diacritic corresponds to a certain phonetic aspect, regardless of the letter modified; (3) each diacritic is just a randomly chosen symbol to differentiate a letter's possible phonemes. This version makes use of all these three principles.

Encoding

PEE is based on Unicode.

Unicode often provides both "combining codepoint" characters that can add diacritics to other characters, and "pre-composed" characters that are letters which already have diacritics. We should prefer the approach which has better rendering in most fonts. For example, the post-composed H̵ (H + U+0335) looks less disturbing than the pre-composed Ħ (U+0126).

Combining codepoints can be found at http://en.wikipedia.org/wiki/Combining_character.

Pre-composed characters can be found by visiting the Wikipedia page for a basic Latin letter (e.g. http://en.wikipedia.org/wiki/A_%28letter%29) and then looking at "Letter <X> with diacritics" at the bottom of the page, where <X> is that basic letter.

All phonetic transcriptions in [...] in this document are in IPA (International Phonetic Alphabet).

The Scheme

1. Unrepresentable or Variable Sounds (UNREP/VAR)

Example: bũsiness

A "~" above (U+0342, which is clearer than U+0303 in some fonts) or below (U+0330) a vowel/consonant letter means this letter's corresponding sound can't be represented by diacritics in this version (because they are rare exceptions to English orthography or are loan words), or the letter's sound is variable depending on context (for example, the "ea" in "read" has various sounds depending on whether "read" is used in the past tense/as a past participle).

Pre-composed characters are preferred if they display better in most fonts.

UNREP/VAR always appears above a vowel letter (ã, ẽ, ĩ, õ, ũ, w̃, ỹ), and usually appears below a consonant letter. If there is not enough space below a consonant letter (e.g. g), it appears above the letter.

UNREP/VAR does not affect vowel/consonant letters around the letter modified.

2. Silences

Example: také

All diacritics for silence do not affect vowel/consonant letters around the letter modified.

2.1. Single-Letter Silence

A "/" above (U+0341, which is clearer than U+0301 in some fonts) or below

(U+0317) a vowel/consonant letter silences this letter.

Pre-composed characters are preferred if they display better in most fonts. An example is í (U+00ED).

The "/" always appears above a vowel letter (á, é, í, ó, ú, ẃ, ý), and usually appears below a consonant letter. If there is not enough space below a consonant letter (e.g. g), it appears above the letter.

A short "-" in (U+0335) a letter can also silence this letter. It serves in cases where we want to avoid excessive separate diacritics. Examples include "how̵", "hey̵" and "doo̵r̵".

Pre-composed characters are preferred if they display better in most fonts. An example is H̵ (H + U+0335).

2.2. Double-Letter Silence

A reverse arch above (U+035D) two letters silences these letters. Examples include right and cheque.

3. Stress (STR)

Example: wikipẹdia

A "." below (U+0323) a vowel letter (ạ, ẹ, ị, ọ, ụ, ẉ, ỵ) means the syllable this vowel letter belongs to is stressed.

Pre-composed characters are preferred if they display better in most fonts. Examples include ị (U+1ECB) and ỵ (U+1EF5).

A multi-syllable word without STR means its stress is variable, e.g. "present".

4. Syllable Separator

Example: a·way

A "·" (U+00B7) between two characters separates two syllables. It is necessary if a word's spelling is not left-associative. For example, "away" should be separated as "a·way" instead of "aw·ay".

5. Vowels

All vowel diacritics affect vowel letters around the letter modified.

5.1. Schwas

5.1.1. Single-Letter Short Schwa

A "\" above (U+0340, which is clearer than U+0300 in some fonts) a vowel letter (à, è, ì, ò, ù, ẁ, ỳ) means that letter, along with vowel letters around it, has a schwa sound (IPA [ə]). An example is "wikipedià".

Pre-composed characters are preferred if they display better in most fonts. An example is ì (U+00EC).

5.1.2. Double-Letter Short Schwa

A "⁓" above (U+0360) "ar", "er", "ir", "or", "ur" and "re" (a͠r, e͠r, i͠r, o͠r, u͠r, r͠e) means it is a schwa sound. An example is worke͠r.

5.1.3. Double-Letter Long Schwa

An arch above (U+0361) "ar", "er", "ir", "or", "ur" and "yr" (a͡r, e͡r, i͡r, o͡r, u͡r, y͡r) means it is a long schwa sound (IPA [əː]). An example is wo͡rk.

5.2. Short Vowels

Without diacritics, the vowel letters a, e, i/y, o and u sound [æ], [e], [i], [ɔ] and [ʌ], e.g. "bat", "bet", "bit"/"gym", "bot" and "but".

Diacritic-equipped vowel letters for short vowels are categorized into four "classes". The first class has the same sounds as the above diacritic-free letters do.

Table of Short Vowel Letters

	a	**e**	**i/y**	**o**	**u**	
1st-Class Short (U+0306)	ă [æ]	ĕ [e]	ĭ/ў [i]	ŏ [ɔ]	ŭ [ʌ]	e.g. băt, bĕt, bĭt/gў̆m, bŏt, bŭt
2nd-Class Short (U+0307)	ȧ [e]	ė [i]		ȯ [ʌ]	u̇ [u]	e.g. ȧny, dėsign, sȯme, pu̇t
3rd-Class Short (U+0311)	â [i]	ê [ɔ]		ô [u]		e.g. privâte, êncore, bôok
4th-Class Short (U+030D)	a̍ [ɔ]					e.g. swa̍p

It is interesting that letters on anti-diagonal lines have the same sound. For example, â, ė and ĭ/ў all sound [i].

5.3. Long Vowels

For convenience, [juː] and [ju] are considered long vowels in this section.

Diacritics-equipped vowel letters for long vowels are also categorized into four classes.

Table of Long Vowel Letters

	a	e	i/y	o	u/w	
1st-Class Long (U+0304)	ā [ei]	ē [iː]	ī/ȳ [ai]	ō [əu]	ū/w̄ [juː]	e.g. tāke, mēet, līght/mȳ, gō, hūge/new
2nd-Class Long (U+0308)	ä [ɑː]	ë [ei]	ï/ÿ [iː]	ö [ɔː]	ü/ẅ [uː]	e.g. cär, ëight, machïne/quaÿ, förce, blüe/jew
3rd-Class Long (U+030F)	ȁ [ɔː]			ȍ [uː]	ȕ [ju]	e.g. tȁll, fȍod, cȕre
4th-Class Long (U+030B)				ő [au]		e.g. rőund

6. R's

All diacritics for "r" do not affect vowel/consonant letters around the letter modified.

The default "r" (without any diacritic) has the [r] sound.

A right-half circle below (U+0339) "r" (ṛ) means this "r" sounds [ər]. An example is expeṛience.

A left-half circle below (U+031C) "r" (r̩) means this "r" sounds [ə]. An example is ouṛ.

A short "-" in "r" (ɾ, U+024D) means this "r" is silenced.

7. Consonants

All consonant diacritics do not affect vowel/consonant letters around the letter modified.

7.1. [tʃ], [ʃ] and [ʒ]

[tʃ], [ʃ] and [ʒ] graphemes are assigned U+032F, U+032E and U+0331:

[tʃ]: ch, ç͟h, t͟, t͟c͡h, t͟ɨ, ç͟, ç͟z, t͡s͟ç͟h
[ʃ]: sh, ş͟h, t͟ɨ, ç͟ɨ, ş͟ş͟ɨ, ş͟ɨ, ş͟ş͟, ç͟h, ş͟, ş͟ç͟ɨ, ç͟é, ş͟ch, ş͟ç͟
[ʒ]: ş͟ɨ, ş͟, z͟, z͟h, t͟ɨ, ş͟h

7.2. X's

The default "x" (without any diacritic) has the [ks] sound.

A right-half circle below (U+0339) "x" (x̩) means this "x" sounds [gz]. An example is ex̩ample.

A "\" below (U+0316) "x" (x̖) means this "x" sounds [kʃ]. An example is anx̖ious.

A left-half circle below (U+031C) "x" (x̜) means this "x" sounds [z]. An example is x̜ylophone.

7.3. Other Consonants

Graphemes for other consonants that may need diacritics are below:

[t]: éḏ
[g]: g, gg, g͡ue, gh
[k]: ç, k, çk, çh, ç͟ç͟, qu, q, çq, çu, q͡ue, kk, kh
[ŋ]: ng, n̩g, n̩, n̩g͡ue, n̩gh
[f]: f, ph, p̀h, ff, f̆h, p̀p̀h
[v]: v, vv, f̩
[θ]: th, t̩h, c͡ht̩h, p͡ht̩h, t̩t̩h
[ð]: t̩h, t̩h
[s]: s, c, ss, sc, st, ꝑs, sc͡h, cc, sé, cé
[z]: ş, z, x̩, zz, ş͟ş͟, zé
[dʒ]: g̀, j, d̩g̀, d̩g̀é, d̩, d̩ɨ, g̀ɨ, g̀é, d̩j, g̀g̀
[w]: u̩