The Role of Morphology in Machine Translation

Bowen Hui

Department of Computer Science
University of Waterloo
Waterloo, Ontario, Canada N2L 3G1
b2hui@uwaterloo.ca

July 1998

Contents

1 Introduction

There is a famous legend that the U.S. government built an English to Russian translation machine that generated the following translations (from Jennings [16]):

English Proverb	Russian Translation	Literal English Translation
The spirit is willing but the flesh is weak.	Spirt khoroshij, a myaso plokhoi.	The liquor is okay but the meat has gone bad.
Out of sight, out of mind.	Nevidno, soshyol s uma.	Invisible idiot.

Table 1: Famous English to Russian Translations

Machine translation (MT) is a widely studied area in Computational Linguistics. Researchers began by studying the processes involved in translating one source language (SL) into one target language (TL) to translating multiple languages. Although MT has been around for a long time, it has not been totally successful. One of the main reasons is due to the fact that not all MT approaches take advantage of the knowledge we have from linguistics. This observation leads to the objective of this paper: to apply various aspects of morphology to MT.

The outline of this paper is as follows. Section 2 provides an overview of three linguistic approaches in machine translation. These methods are direct, transfer, and interlingua. A summary with a comparison of the three approaches is given. We observe that many problems arise from lack of morphological theories, which leads to the discussions in section 3 and 4. Section 3 briefly introduces various aspects of morphology in linguistics. It reviews different types of word formation processes, such as derivation, inflection, compounding, clipping, and blending, as well as motivating the need for a two-level morphological theory for accurate performance. Section 4 discusses four major MT problems. These problems are stemming, lexical ambiguity, the structure of the lexicon, and lexical choice. We explain how these problems affect MT and how some solutions can be aided with morphological analysis. Having studied the theories behind linguistics and machine translation, section 5 outlines a program that compares different techniques in automatic stemming and affixation. Finally, section 6 gives a summary of this paper and points out directions for future work.

The reader should be aware that non-linguistic approaches have been taken as well. Introductory books by Hutchins & Somers [15] and Arnold *et al.* [2] give a good description for some of the approaches which are not discusses in this paper. These approaches include pure statistically-based approaches (Brown *et al.* [5]), the *Shake-and-Bake* approach (Beaven [4]), knowledge-based approaches (Goodman & Nirenburg [8], Goodman [7]), and example-based approaches (Nagao [19], Somers [26]).

2 Linguistic Approaches to MT

Three main linguistic approaches to machine translation are surveyed, namely *direct*, *transfer*, and *interlingua*. The objective of this section is to give an overview of what these linguistic methods are, how they work, and what their advantages and disadvantages are. The main conclusion we draw at the end of the section is that none of these approaches give satisfactory performance for cross-linguistic translations.

2.1 Direct

Direct translation is one of the early approaches to machine translation. This method is viewed as a pattern matching approach, because the idea behind it is to take each word from the source language and replace it with its "equivalent" in the target language. Some systems may also apply simple reordering rules before generating the translation in the target language, but the analysis relies mainly on morphology. For example, suppose we want to translate 'Mary likes John' into Spanish. We would type the sentence in as the input and expect 'Juan gusta Maria' as the output. But what goes on in between these two steps? First of all, the input goes through a morphological analysis stage, in which processes such as stemming[1] takes place. In this example, 'likes' is stemmed into its root form, 'like', and 'Mary' and 'John' remain the same (depending on the sophistication of the system, the analysis may extract information such as person, gender, and number for 'Mary' and 'John'). The next step is to look up these words in an English-to-Spanish dictionary, and replace these words by 'Maria', 'gusta', and 'Juan' respectively. Finally, if the system has a set of local reordering rules available for English to Spanish, then the subject and object would switch places, resulting in 'Juan gusta Maria', which is the expected output. Figure 1 below illustrates the modules of the processes just described.

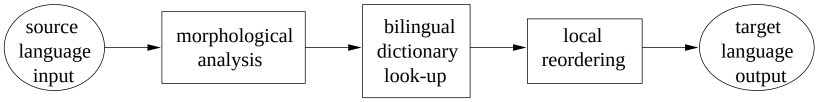

Figure 1: Direct MT System

The analytical processes in the direct approach include identifying inflections, reducing inflected forms to root forms, storing the results into a look-up table, and possibly applying some local reordering rules. Some systems do not have reordering rules because it is too tedious for the designer or programmer to list *all* the rules as well as exceptions between the two languages. Direct translation depends strongly on a direct mapping of the words from SL to TL. Although some minimal reordering rules may apply during the translation, these rules are linear (i.e., they are defined in terms of adjacent words) and hence many problems arise. For example, 'college junior' and 'junior college' have different meanings. To differentiate between these two phrases, a direct system would have to recognize that when 'college' precedes 'junior', the meaning is "a student who is at a junior level", and when 'junior' precedes 'college', the meaning is "a college for junior students". However, this linear rule is broken by the phrase 'junior in college', which has 'junior' before 'college' but the meaning is still "a student who is at a junior level". The reader can imagine how tedious and unrealistic it is to list out linear rules soundly and completely.

Free rides are a big help to direct translation systems. The notion of free rides is basically getting what you want without paying for it. In the context of MT, free rides would come into play when the system is translating between languages that have similar morphology and syntax. This way, even if the system is not sophisticated, it is still likely for the system to generate the correct output because the similarity of SL and TL have reduced the need for a deeper understanding and analysis of the languages. This is not to say that there is no effort involved in translation using the direct method. It is simply that linearity limits what can be achieved to the extent that doing a lot of work results in very little additional results. Therefore, the best use of direct translation systems is as a preprocessor that supplies the translated output for human translators to do further editing.

2.2 Transfer

The inaccuracy of the direct translation mechanism led to the notion of an intermediate representation of meaning. This idea led to a new approach, called *transfer*. Although semantics is incorporated into this approach, the representation is still language dependent. Some researchers (Dorr [6]) view the transfer approach as rule-based, because the transfer modules consist of unidirectional rules that are specifically designed for a particular pair of languages. In other words, if a transfer system is built for translation from English to Japanese and if the designers and programmers now want to build a system that translates Japanese to English, then they would basically have to start from scratch. The reason behind this will become more clear by examining the architecture of a transfer system shown in figure 2.

From the architecture, we see that both SL and TL require their own analysis module as well as their own generation module. This entails that given SL, we need to analyze it, apply language dependent rules, and generate TL. Therefore, a system that translates English into Japanese has low reusability for a system that translates Japanese into English, because all of the modules are language and task specific and unidirectional.

Consider the earlier example and let us examine how transfer applies. The input to the system is 'Mary likes John', which gets passed on to the SL analysis module. Within this module, the main tasks are morphological analysis and syntactic analysis (which the direct method does not have). Morphological

[1] Stemming is a morphological process that identifies words with the same root form as one concept. The details of this process is discussed further in section 4.1.

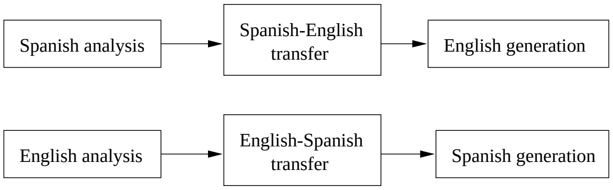

Figure 2: Transfer MT System

processes are similar to the ones performed in direct, while syntactic processes include constructing a parse tree and extracting simple semantic information[2]. After the analysis process is complete, we may obtain a parse tree specific to the English sentence 'Mary likes John' with semantic features and lexical items in it. This tree is sent to the $SL \rightarrow TL$ transfer module for applying any necessary language specific rules such as reordering. In our case, the rule for swapping the agent[3] and the patient[4] of the sentence is applied to the tree. (If the translation requires adding a preposition in front of the patient for grammatical reasons, then the patient noun phrase would be changed into a prepositional phrase in this step. This analysis suggests that a deeper representation is needed so to avoid having to deal with such details.) Then this tree goes into the TL generation module, in which lexical mappings in the target language takes place. At this point, we get the mappings 'John' to 'Juan', 'likes' to 'gusta', and 'Mary' to 'Maria' . The tree is compressed into a sentence, which becomes the output in the target language.

Because this is a simple translation example, it may be hard to see why the transfer method is better than the direct method. Let us consider a translation from Turkish to English (data taken from O'Grady [21])[5].

(1) Adam evi Ahmede gösterdi.
(2) Adam-∅ ev-i Ahmed-e göster-di.
(3) man-NOM house-ACC Ahmed-DAT show-PAST
(4) The man showed the house to Ahmed.

The sentence in (1) is a Turkish sentence with a morpheme breakdown shown in (2). Sentence (3) shows a morpheme-by-morpheme translation into English. Capitalized letters are syntactic markers which will be discussed shortly. Lastly, (4) is the English translation of (1).

Suppose we are trying to translate the Turkish sentence in (1) into English using the direct method. If we can analyze the morphology of Turkish successfully, we would be able to strip the words to their stems, which are 'adam', 'ev', 'Ahmed', and 'göster'. Then we would look up these words in our bilingual look-up dictionary and map them (assuming successfully) to 'man', 'house', 'Ahmed', and 'show' respectively. At this point, we have a list of words without knowing how one relates to the other. We may be lucky and magically end up with 'the man showed the house to Ahmed'. Or we may end up with 'Ahmed showed the house to the man' since we have lost the case markers. However, with the transfer method, we can analyze case features from our parse tree and store this information so that the English translation reflects the correct relationships among the words.

Although this is an improvement over the direct approach, the transfer method is still language specific (for example, the parse trees have lexical items attached and rules are unidirectionally based on the syntax of the particular source and target languages). Due to this dependency, transfer systems are most applicable to languages that have similar linguistic features because the accuracy of the translation depends on the rules and representation of the transfer modules.

[2] An example of semantic information that transfer requires is *case*. Case is "a category that encodes information about an element's grammatical role (subject, direct object, and so on). (O'Grady [21])" English pronouns exhibit case for *nominative* (I, they, he), *accusative* (me, them, his), and *genitive* (my, their, his).

[3] An *agent* is the thing (animal, object, etc.) that initiates an action. It is often referred to as the *subject* of a verb.

[4] A *patient* is the thing (animal, object, etc.) that undergoes an action. It is often referred to as the *object* of a verb.

[5] Nominative (NOM) marks the agent; accusative (ACC) marks the direct object; dative (DAT) marks the recipient; past tense (PAST) marks an action in the past.

2.3 Interlingua

Interlingua is an idealistic approach to solving the machine translation problem because it relies on conceptual representation and linguistic principles. Unlike transfer, the intermediate representation is language independent. This feature makes interlingua theoretically most attractive because it is the closest technique to arriving at a universal solution. Figure 3 below illustrates the architecture of an interlingua system.

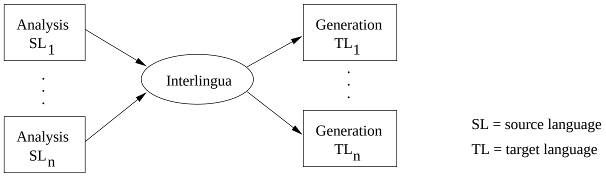

Figure 3: Interlingua MT System

This architecture is similar to that of transfer, except that there is only one module that serves as the intermediate representation. With interlingua, if we have a system that translates from English to Japanese and we want it to translate from Japanese to English, then all we have to do is just add one Japanese analysis module and one English generation module.

Again, if we go back to the example with 'Mary likes John', we can see more clearly how interlingua is different from transfer. The first step is the same as transfer, where the input sentence gets processed in the SL analysis module. The kind of analysis that is done here is similar to that of transfer, with the addition of semantic extraction. The analysis module outputs enough information for the interlingua module so that it can form abstract *concepts* for the sentence and construct the relationship among these concepts. In our case, concepts would be created for 'Mary', 'like', and 'John', which can be roughly described as a named female human, a state of positive feelings, and a named male human respectively. The relationship among these concepts would be that the named female individual is in a state of having positive feelings for a named male human. A pictoral representation of these concepts and their relationships is shown in figure 4 (where a state is represented as a square, a concept as a circle, and the relationships as directed links).

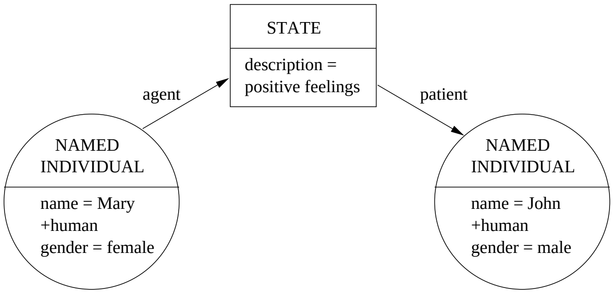

Figure 4: A Representation for 'Mary likes John'

After the concepts have been built, the system can generate the sentence in TL by performing the appropriate lexical and syntactic processes. At this point, we get 'Maria' as the named female human, 'gusta' as the state of positive feelings, and 'Juan' as the named male human. The major difference in the interlingua approach is that no "swapping rules" exist. Instead, the TL generation module will generate the correct sentence based on the syntax of TL only. So it will see that the predicate 'gusta' requires a patient and an agent (in that order) and arrange the concepts (and consequently their lexical equivalences) in that fashion. This procedure shows that the generation process requires no knowledge of SL or the original input sentence. The final output of the translation is 'Juan gusta Maria'.

Although interlingua purports to be a universal model, the core of its problems are due to the fact that linguistic universals are incomplete. The main problem in interlingua systems is representing crosslanguage concepts. Just how should a set of universal concepts (in the interlingua module) be defined? We need to determine exactly what constitutes a concept and how to extract the necessary information we need from the texts. Consequences in poor design may result in choosing a representation that is not language neutral and encoding excessive and unnecessary information. For example, the word 'rice' has six[6] different interpretations of rice in Malay. If we want to translate 'we are having rice for dinner', we obviously mean 'cooked rice'. However, if we do not encode this information, we may end up choosing 'uncooked rice' or 'unharvested grain' instead. This example illustrates that if the interlingua system has an inadequate representation of concepts then the accuracy in translation can be greatly affected. Since these problems are hard to solve, interlingua systems incorporate contextual and real world knowledge in practice. This information can help resolve ambiguity problems that may arise.

2.4 Summary

In this section, we surveyed three linguistic approaches in MT: direct, transfer, and interlingua. Direct translation is a naive method that maps words in SL directly to words in TL with the help of some local reordering rules. Transfer is an indirect approach that supplemented the direct method with an intermediate layer of representation. This addition brought along syntactic and semantic knowledge that encodes language specific rules to the system. Interlingua is a theoretically appealing approach to MT that translates language neutral, or universal, concepts rather than actual words and phrases. Both the direct and the transfer approach are more suited for bilingual translation (i.e. translation between two languages) while the interlingua approach is better for multilingual translation (i.e. translation among n languages, where $n > 2$). Whether we are designing a bilingual or multilingual system, the system should encode language specific information for better precision as well as language neutral information for simpler modeling.

All of these approaches have advantages and disadvantages. The direct method is easy to implement for bilingual translation because the basic idea is to look up each word in the bilingual dictionary and select a corresponding translation. However, the accuracy is low, because of ambiguity in word meanings and the lack of syntactic knowledge in the system. Furthermore, it is impratical to build a multilingual system based on the direct method, because adding the n^{th} language requires adding $2(n-1)$ mapping rules (Naruedomkul & Cercone [20]). The general conclusion we make of the direct approach is that higher linguistic information (such as syntax and semantics) needs to be represented in the system for obtaining better results. This leaves us with the transfer and interlingua approaches.

The transfer method trades off accuracy with the amount of language specific rules encoded in the system. Because all the rules are defined in terms of a pair of languages, a multilingual system would require n analysis modules, n generation modules, and n^2 unidirectional transfer modules. On the other hand, the interlingua approach is designed for multilingual translation. For n languages, the system needs n analysis modules, n generation modules, and *one* intermediate representation module. Although there are considerably fewer intermediate modules in interlingual systems, the analysis and generation modules in transfer systems are much simpler. Interlingua modules need to encode the same level of detail of information despite the similarity of languages that are being translated. Since interlingua strives for universality, the analysis and generation modules need to encode very specific details *all the time*. This is not the case with transfer systems. Transfer modules encode information at the level of detail that is required based on two languages (SL and TL); the more similar SL and TL are, the less detail is needed.

From these three approaches, we conclude that none of them are satisfactory both in theory and in practice. The aim of this paper is not to come up with a new and better approach, but to search for ways in which morphological theories can improve the performance in the existing approaches. We saw with the Turkish example that it is important to analyze words into their parts correctly so we can identify features such as case. We also saw with the Malay example that there is more than one sense for the

[6] The six translations are:

Malay	**English Gloss**
padi	'unharvested grain'
beras	'uncooked'
nasi	'cooked'
emping	'mashed'
pulut	'glutinous'
bubor	'cooked as a gruel'

This is data is taken from Hutchins & Somers [15].

English word 'rice'. In the following sections, we will see how morphology can give us insight into these problems. Last but not least, morphology will play a large role on how an MT system should structure its dictionary.

3 An Overview of Morphology

In the previous section we surveyed three linguistic approaches to machine translation. However, it was not always clear where linguistics come into play or how linguistics can improve MT accuracy. In this section, we will give an overview of *morphology*, a module in linguistics that explain how words can be broken up into smaller components and how words can combine to form other words. With this knowledge, we can then go on to solving some MT problems in the next section.

3.1 Types of Morphological Processes

The major types of morphological processes are *derivation* and *inflection*. Derivation is usually associated with affixation and the change of syntactic category while inflection is usually associated with the marking of grammatical information. Another phenomenally productive process is *compounding*. Our discussion here also examines nominal compounds. Lastly, we will briefly look at *clipping* and *blending*, which are common processes in many languages. This section will provide a self-contained introduction to these morphological processes. There are other word formation processes as well, such as acronyms and onomatopoeias, but they are not discussed here.

3.1.1 Derivation and Inflection

One of the most common word formation processes is *derivation*. For example, the word 'penniless' is derived from the word 'penny' plus the suffix `-less`[7]. This process usually changes the syntactic category to which the root belongs and it tends to preserve the meaning of the root. In our example, 'penny' is a noun and the resulting category is an adjective. The root 'penny' has the meaning of "a coin, one-hundredth of a (Canadian) dollar", and the suffix `-less` means "without" or "lack of". The output meaning of 'penniless' is thus "without any money".

More than one affix[8] can be attached onto the same root, as in 'unhappiness'. In this case, we have two possible word internal representations (see figure 5). One representation has `un-` attaching onto 'happy' before `-ness` is attached, while the other has 'happy' attached to `-ness` before `un-` is attached. Although the distinction between these two representations are subtle, they are important if we want to extract an accurate meaning from the word. One way to find the correct representation is to examine the behaviour of the affixes with other words. By observation, `un-` only attaches onto adjectives (i.e. not nouns), so we can rule out the second (right) representation and conclude that the first (left) one shows the correct analysis.

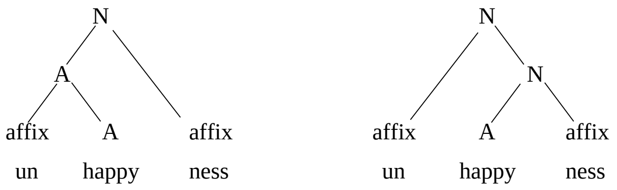

Figure 5: Possible Structures for 'unhappiness'

Special instances of derivations are *conversion* and *backformation*. Conversion, also called *zero affixation*, converts a word into another syntactic category without affixation. For example, 'empty' is an

[7] The convention used to show an affix boundary is by a dash '`-`' or a plus '`+`' sign. Some authors also use a hash symbol '`#`' to differentiate types of affixes.

[8] Affixes are a type of *morphemes*, which is the smallest meaningful unit in a language. Examples of morphemes are the regular plural marker in English `-s` or `-es` and common stems such as `berry` in 'rasberry', 'cranberry', and 'strawberry'. Note that the English plural can take more than one form. This is called *allomorphy*.

adjective that has been derived into a verb, 'ship' is a noun that has been derived into a verb, and 'permit' is a verb that has been derived into a noun. Both 'empty' and 'ship' preserve the original phonological properties while 'permit' results in a change of primary stress. Backformation is a process where words with common affixes are the underlying representation as opposed to the resultant. For example, 'oriental' has a common suffix `-al`, so speakers assume that the noun to which `-al` attaches also exists. Here, 'orient' is derived. Other examples include 'enthuse' from 'enthusiasm' and 'edit' from 'editor'.

Another common morphological process is *inflection*, which is a process that builds grammatical information onto words. Inflection can be used to identify tense, person, gender, case, number, and noun class, although not every language encompasses all of these grammatical markers. An example of inflection is 'apples', where the plural `-s` indicates the number. This process is called affixation, as we have seen earlier. There are other processes that marks inflection as well, such as suppletion ('swam' from 'swim') and reduplication (*tatabuh* 'will run' from *takbuh* 'run' in Tagalog).

3.1.2 Compounding

Compounds result from concatenating existing words together to form new meanings. In English, nouns, verbs, adjectives, and prepositions can concatenate to form new words. In particular, English allows the following constructions (from O'Grady [21]):

Rules	**Examples**
$N + N \rightarrow N$	streetlight
$A + N \rightarrow N$	bluebird
$P + N \rightarrow N$	in-group
$V + N \rightarrow N$	washcloth
$N + V \rightarrow V$	spoonfeed
$A + V \rightarrow V$	whitewash
$P + V \rightarrow V$	underestimate
$V + V \rightarrow V$	break dance
$N + A \rightarrow A$	sky blue
$A + A \rightarrow A$	deep blue
$P + A \rightarrow A$	over ripe

Table 2: A list of English Compounds

From table 2, we notice that the rule $V + A$ is missing and that prepositions cannot be generated from compounds (i.e., $*X + P \rightarrow P$, where X is any category). This set of examples shows the productivity of compounding in English.

In general, there are two main types of compounds: *endocentric* and *exocentric*. Endocentric compounds have their meanings derived from the rightmost unit. For example, 'dog food' is a type of food and 'food' is the rightmost unit. (We say *unit* and not *word* because we can get larger compounds such as 'dog food eater'.) On the other hand, exocentric compounds have their meanings derived from the leftmost unit. For example, 'red head' is not a kind of head, but a person with red hair.

One of the most discussed problems is nominalization of compounds. This type of compounds is sometimes referred to as "noun-noun compounds" because the process takes an input of two nouns and generates another noun as the output. An illustrative example is 'table tennis', where 'table' and 'tennis' are both nouns, and they together form another noun. The meaning of 'table tennis' is roughly "a sport similar to tennis but the setting is on a table". Furthermore, we can generate 'table tennis tournament' by inputting 'table tennis' and 'tournament' in that order. The meaning behind this new noun is basically "a tournament for the sport, table tennis". This example shows us how the input nouns play a role in the meaning of the output noun. However, the hardest part of the nominalization problem is not to determine the meaning of such compounds (because in most cases, the meaning is derived from its parts) but to identify what these parts are. Let us consider an example from German (taken from Hutchins & Somers [15]). The German word "Alleinvernehmen" can be decomposed as *All* and *Einvernehmen*, which means 'global agreement' or as *Allein* and *Vernehmen*, which means 'lone perception'. These two meanings are quite distinct, so a random guess at the decomposition would not suffice.

3.1.3 Clipping and Blending

Other morphological processes used to create new words include clipping and blending. Clipping takes a polysyllabic word and removes one or more of its syllables. Common examples are ‘prof’ from ‘professor’, ‘info’ from ‘information’, and ‘demo’ from ‘demonstration’. Blending takes multiple words (usually two) and combine parts of it into one word. Examples are ‘brunch’ from ‘breakfast’ and ‘lunch’, ‘Nortel’ from ‘Northern’ and ‘Telecom’, and ‘bit’ from ‘binary’ and ‘digit’.

3.2 The Need for A Two-level Morphology

We have discussed a number of morphological processes, yet we have not considered how to order them. Consider the following paradigm (taken from Hui [14]):

Adjective	**Nominal**	`+ity`	`+ness`
glorious	glory	*gloriousity	gloriousness
spacious	space	*spaciousity	spaciousness
pious	*	piety	piousness
tenacious	*	tenacity	tenaciousness

Table 3: Productivity of +ous, +ity, and +ness

The pattern observed here is that nominals and the process `+ity` block each other. Where there is an existing noun for the corresponding adjective, the process `+ity` cannot be applied; where the process `+ity` is applied, the nominal does not exist. How can we explain this? Furthermore, how can we explain that the process `+ness` can be added in all the cases?

Obviously, not all words are suitable candidates for all processes. Is this because some words do not satisfy the conditions of the processes (e.g., `+ism` does not attach to ‘play’ because the suffix expects a noun or an adjective)? Or is it because certain rules are ordered before others, i.e., certain rules are *blocked*? In fact, both of these factors have a part in the productivity of word formations. Conditions are needed as a way to rule out invalid candidates *locally* and an overall schema of ordering is needed to structure these rules *globally*.

Linguistically, there are two levels of lexical morphology and phonology. Level one of morphology contains less productive morphemes and level two contains more productive morphemes. The process of an underived word (w_1) would be passed into level one of morphology, then to level one of phonology, and cycles back to level one of morphology. The output of level one morphology is a word (w_2) that is passed to level two morphology. Again, it cycles to level two phonology and back to level two morphology. The final output is a derived word (w_3). A graphical representation of this process is shown in figure 6 below.

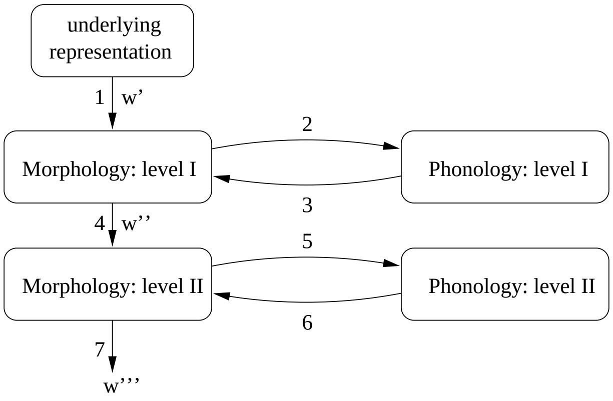

Figure 6: Interaction in Two-level Morphology

An example to illustrate lexical morphology follows. The underlying verb 'protést'[9], meaning "the action of one person objecting something", enters level one of morphology. It undergoes the process $V \rightarrow N$ and becomes a noun. The meaning of this word has changed to "the act of one person objecting to something". Then it enters level one of phonology, where the stress rule for zero affixation applies, and the noun becomes 'prótest'. It then goes back to the morphology module and enters level two of morphology as a noun. In the second level, the process $N \rightarrow V$ applies, and the noun 'prótest' becomes the verb 'prótest', meaning "the action of a group of people objecting something". This word goes to level two of phonology, finds that no rules apply, and goes back to morphology and exits the derivation process. The result is the verb 'prótest'.

Why do we need two levels of morphology and phonology? Consider the following paradigm:

V	N	N
guide	guide	*guider
spy	spy	*spier
cook	cook	cooker
divide	divide	divider

Table 4: Productivity of the Agentive +er

We want to explain when the agentive +er can be added to verbs given that the agentive process is on level two of morphology[10]. Table 5[11] below shows the rules we are concerned with.

level one	level two
$V \longrightarrow N$	$N \longrightarrow V$
$A + ity$	V+er (agentive)

Table 5: Some Rules in Level One and Level Two Morphology

The basic idea is to treat 'guide' and 'spy' as one class and 'cook' and 'divide' as another class. If we hypothesize that these two classes have different underlying representations then we can get the correct derivations. For example, suppose 'guide'[12] is underlyingly a noun. It enters level one morphology and level one phonology, and nothing applies. Then it enters level two morphology. The agentive rule does not apply because 'guide' is a noun. But $N \rightarrow V$ applies, so we get 'guide' as a verb. At this point, we can apply 'guide' + er $\longrightarrow$ *guider, which is the wrong result.

Suppose 'guide' is underlyingly a verb. It enters level one of morphology, undergoes the process $V \rightarrow N$, and exits level one as a noun, 'guide'. Rules in level two morphology do not apply, namely 'guide' does not undergo the agentive process because the process only applies to verbs.

Now consider 'cook'[13]. If 'cook' is underlyingly a verb, then it enters level one morphology, becomes a noun, enters level two morphology, and nothing applies. Therefore, we cannot derive 'cooker' in level two. However, if 'cook' is underlyingly a noun, then it enters level one morphology, nothing applies, and enters level two morphology. In this module, both 'cook' (as a verb) and 'cooker' are derived. This analysis is consistent with the data.

3.3 Summary

We have studied various aspects of morphology from a theoretical linguistics point of view. Specific areas of morphology we examined include derivation, inflection, compounding, clipping, blending, and lexical morphology. However, it is not always clear how this knowledge can help us in machine translation. Does

[9] The notation $\acute{v}$ means that the vowel v has primary stress.

[10] The justification behind which rule belongs to which level is beyond the scope of this paper.

[11] Table 5 only shows four rules which are needed for the current example. Other rules for level one morphology in English include the suffixation of +ity, +y, +ive, +ize, and +ion and rules for level two morphology include the suffixation of +ness, +less, +ful, +ly, and +ish.

[12] The same argument applies to 'spy'.

[13] The same argument holds for 'divide'.

it matter whether a word has several word internal representations? How can we determine whether a word is a noun or a verb? The answers to these questions lie in the next section.

4 Morphologically Related Problems

This section surveys four major morphological problems in MT. First, we study a process called *stemming* by looking at some English examples and describing the Porter stemmer[24][14]. Then we turn to *lexical ambiguity*, which is a very hard problem for natural language processing. We present the problems that arise in three types of lexical ambiguity followed by partial solutions using the morphological knowledge from the previous section to solve these problems. The third part of this section discusses the structure of the lexicon and the kind of information it stores. This discussion draws together research from machine translation and psycholinguistics because we want the machine lexicon to be closely related to the human lexicon in structure. In the fourth part, we will briefly discuss the importance of lexical choice as a general natural language generation problem. Finally, a summary of this section is provided.

4.1 Stemming

Stemming is a process that removes morphemes from a word such that the result is an underived word. For example, 'employer', 'employee', and 'employment' have the common root 'employ'. So stemming these words means removing the suffixes `-er`, `-ee`, and `-ment` respectively. These words are now trimmed to the stem 'employ'. Why might such a process be useful in MT? The reason is we want to identify the root form of words so we can make correct mappings to roots in the target language. However, the reader should keep in mind that the drawback in stemming is that it introduces ambiguity. In this section, we only consider stemming in English[15].

To appreciate the problem, consider the example of stripping the English suffix `-ing`. Obviously, we only want to strip off `-ing` if it indicates the progressive tense of a verb. For words such as "playing", "singing", and "walking", the process is quite straightforward. What about "dying", "betting', and "babbling"? These verbs require an additional step in the analysis so that the end result turns out as *die*, *bet*, and *babble* respectively. For "dying", we can hypothesize some kind of "*reverse* y-replacement" rule, where we substitute the `-y` with an `-i` or `-ie` because these words are often replaced by a `y` during an affixation process. For "betting", we can hypothesize some kind of "reduce gemination" rule, where double consonants (geminates) are reduced to single consonants. Finally, for "babbling", we can hypothesize some kind of "add `-e`" rule, where an `-e` is attached onto certain words. But how do we differentiate "die", "bet", and "babble" from "play", "sing", and "walk"? Furthermore, the stripping analysis would require the program to recognize words such as "sling", "fling", and "sing" as root forms so they do not get stemmed to "sl", "fl", and "s" respectively.

One of the most popular stemming algorithms is the Porter algorithm. The algorithm attempts to identify words that have common roots. For example, 'connects', 'connected', and 'connecting' all get stemmed to 'connect'. The Porter stemmer removes predefined suffixes in five successive steps (refer to appendix B for the list of steps). The algorithm is careful not to remove suffixes that will result in stems without vowels (e.g., removing 'ing' from 'sling'). However, it does not consult a dictionary during the suffixation process which means that some stems do not end up as orthographic words. For example, the Porter stemmer will stem 'move', 'moving', and 'moved' into 'mov', but 'mov' is not a word in English. This can cause problems when the MT system tries to find the corresponding word for 'mov' in the target language. If the system uses such an algorithm, it would either not find a correct translation for these words, or it will have to store these words into the dictionary as well. Perhaps the algorithm can be modified so that it consults the dictionary to see if the stemmed word is an actual word or not. Furthermore, these algorithms are not linguistically based, so theories in morphology have no impact on

[14] The author does not know what level of sophistication the stemming algorithms are in machine translation. However, the Porter algorithm is mentioned so we can understand what goes on in the stemming process better.

[15] The reader should note that there are many languages that have richer morphology than English. Stemming would have a greater impact on languages that have more consistent orthography and a richer morphology than English. For example, studies in information retrieval showed that suffix stripping does not affect English text retrieval significantly (Harman [11]) while very positive results were found in Slovene text retrieval (Popovic & Willet [23]).

the underlying structure of the algorithms. Morphological analyzers based on *two-level morphology* exist but their applicability and effectiveness in MT are beyond the scope of this discussion.

4.2 Lexical Ambiguity

Ambiguity occurs when more than one meaning can be interpreted. *Lexical ambiguity* is a type of ambiguity caused by some aspect of morphology. In most cases, the underlying cause for lexical ambiguity is *lexical gaps*, that is, when there is no one-to-one correspondence between the words in SL and the words in TL. Another cause is analyzing words in isolation rather analyzing words in context. Attempts at solving this problem have been made by collecting statistical information on the frequency of adjacent words. Although this approach is helpful, words that occur together do not always have the same meaning. Recall from section 2.1 the difference in meaning between 'college junior' and 'junior in college'. This simple example already illustrates the limitations of statistical approaches. What we need to solve ambiguity is to use information from other aspects of language. For example, we can use morphology to solve a syntacticly ambiguous problem, and we can use syntax to solve a morphologically ambiguous problem. This is the point of view we take here.

The present section surveys three kinds of lexical ambiguity and the hardship they cause in MT. These three types are: (i) categorial ambiguity, (ii) homographs, homophones, and polysemes, and (iii) translational ambiguity. Following that discussion is an original approach to translational ambiguity proposed by the author. Finally, we examine some morphological clues to recognize compounds.

4.2.1 Categorial Ambiguity

A type of syntactic ambiguity is *categorial ambiguity.* Consider the following sentence taken from Hutchins & Somers [15]: "Gas pump prices rose last time oil stocks fell". It is very striking that humans do not find this sentence ambiguous even though every word belongs to at least two syntactic categories (such as noun and verb, or adjective, verb, and noun). Machines have a hard time overcoming categorial ambiguity because most parsers are not sophisticated enough. If a parser attempts to determine a precise and perfect parse for an ambiguous sentence like the one above, then it would have to try every possible combination of the words given their possible syntactic categories. This method seems like a waste of time when we reflect on how easy it is for a human reader to interpret the sentence "Gas pump prices rose last time oil stocks fell". Another solution is to apply morphological knowledge to this problem. In particular, we can identify the internal structures of the words in the sentence, and use the affixes as clues to help us identify the syntactic category of the word. For example, if a word has a suffix `-ing`, then we can hypothesize that it is a verb in the progressive tense. However, this approach is not always deterministic. If a word has a suffix `-s`, then we can hypothesize that it is a verb with a third person singular subject or a plural noun, but additional information is needed to resolve the ambiguity between these two choices. Although morphology cannot solve categorial ambiguity completely, it is one step towards a solution to this syntax problem. We conclude here that categorial ambiguity remains to be a large barrier for syntactic parsers.

4.2.2 Homographs, Homophones, and Polysemes

Homographs, homophones, and polysemes are three types of ambiguity that arise in a monolingual context. *Homographs* are words that are spelled the same way but with different meanings. For example, the spelling "pen" can mean a writing impliment, a fenced area for domestic animals, a small play area for children, or a bomb-proof shelter for submarines. *Homophones* are words that have the same pronunciation but with different meanings. For example, "two" and "too" are homophones. *Polysemes* are words that have a large number of similar meanings. For example, the word "mouth" can refer to the mouth of a river, a human, or a musical instrument. All of these have similar meanings but they refer to different objects depending on the context.

It is important to solve these problems in MT because we want an accurate translation. Even though homographs, homophones, and polysemes are monolingual phenomena, they will cause problems during the translation process. For homographs, if a language uses different spellings for the English spelling of "pen", then how do we determine which spelling is the correct one? For homophones (specific to speech translation only), how do we know if a pronunciation is referring to the number "two" or the agreement "too"? For polysemes, if a language uses different words for the various senses of "mouth", then how do we choose the correct sense? It is easy to imagine many more situations where these cases are problematic in MT. The problem of polysemy is developed further in the following subsections.

4.2.3 Translational Ambiguity

Translational ambiguity occurs when a word or phrase from SL is unambiguous to a native speaker of SL, but it can potentially translate to more than one word or phrase in TL. This type of ambiguity is important to the topic of MT because not only do we want to arrive at a grammatically correct translated output, but we also want to get a translation that preserves the meaning of the original input.

There are several types of ambiguities that arise during translation. The first is stylistic translational ambiguity, which occurs when the translated text type (such as a newspaper article or a children's story) conditions the lexical choice. An example would be the degree of killing, which can be expressed as killed, assassinated, murdered, died, stabbed to death, etc. Obviously, a less graphic word would be chosen for a children's story, while a more informative but objective word would be chosen for a newspaper article.

Another type of translational ambiguity is grammatical translational ambiguity, which occurs when the grammatical context conditions the lexical choice. An example is the translation of the verb 'to know' into French (taken from Hutchins & Somers [15]). Depending on the sentence, a native speaker of French may find *connaître* to be a more natural translation than *savior*. Examples of these are '*Je connais la bonne réponse*' versus '*Je sais quelle est la bonne réponse*'.

The third type of ambiguity that arises during translation is called conceptual translational ambiguity. This occurs when one word in SL corresponds to more than one concept in TL. Recall the example from section 2.4 (cf. footnote 6) that the word "rice" corresponds to six different concepts of rice in Malay. Without knowledge of these six concepts, there is no guarantee that the translation is correct.

One of the major causes of these ambiguities is due to lexical gaps, where certain lexical items are present in one language and are absent in others. Common examples of lexical gaps are found in colour spectrums from different cultures. For example, the native language Sliammon does not have a word for *purple*, because the speakers describe it as a type of blue. This is analogous to human communication, where each human has a distinct lexicon. So during communication, one person may say something that another participant in the conversation would not know. In this case, the second participate will ask the first to repeat and explain the word, so s/he can learn a new lexical item. In a similar way, this is what machines need to do when they encounter new words.

4.2.4 Using A Word Hierarchy

Hutchins & Somers [15] described a number of words that correspond to more than one concept in another language. We already saw that there are six senses of rice in Malay. Another example is the word 'wear' (where the meaning is 'to have on an item of clothing or accessory'). In English, the same verb is used to describe what we have on, whether it be a hat, a belt, or a shoe. However, in Japanese, the word to express this meaning depends on the item that is described. Hutchins & Somers [15] gives eight senses of 'wear' in Japanese:

(1) *kiru* (generic)

(2) *haoru* (coat, jacket)

(3) *haku* (shoes, trousers)

(4) *kaburu* (hat)

(5) *hameru* (ring, gloves)

(6) *shimeru* (belt, tie, scarf)

(7) *tsukeru* (brooch, clip)

(8) *kakeru* (glasses, necklace)

This kind of lexical gap is an instance of translational ambiguity. To tackle this problem, the author hypothesized a hierarchical approach to classify ambiguous words. To demostrate this approach, the author devised an experiment with English sentences involving all sorts of "wearable" garments. These sentences were given to native speakers of Japanese and Cantonese so that we determine the appropriate lexical term that is used for different wearable items. From there, we hope to classify the different terms used into categories based on the kind of wearable garments. An example of a possible category is "items worn on the hands". Two advantages follow if such categories exist. One advantage is that we can develop a general treatment of words using word hierarchies. Another advantage is that this hypothesis can predict which lexical entry should be used for newly created garments.

This hypothesis was tested with a list of original English sentences[16] using the progressive form of the verb 'to wear'. These sentences were translated into Japanese and Cantonese (the dialect spoken in Hong Kong) by two native speakers. No comparison is made between the translation of 'to wear' and 'to put on', or any other semantic variants of 'wear'. Sentences used in the experiment involve the meaning of wearing something on a human body. Therefore, other senses such as the one intended in "the magic is wearing off" are not tested. As mentioned above, Hutchins & Somers listed eight Japanese words for 'wear'. However, in this experiment, 'shimeru', 'kakeru', and 'haoru' are not used. The author consulted the speaker on this matter and the gap is caused by the difference between formal speech and colloquial speech. This point should be kept in mind during the stege of data analysis. Although this data sample is small, it will suffice as a preliminary experiment to test the validity of the hypothesis.

Tables 6 and 7 below summarize the results of the data collected in the experiment. In this paper, the Japanese data is recorded in Romanji. Since Cantonese does not have a standard alphabet, we use the International Phonetic Alphabet (IPA)[17] to transcribe the data. The transcription provided here do not include tonal markings. These tables show that four Cantonese verbs (*taɪ, ta, t͡sʰa, t͡sœk*) and seven Japanese verbs (*hameru, kaburu, suru, tsukeru, nuru, haku, kiru*) are translations of 'wear' in English. Asterisks on a wearable item denote multiple verb usage.

taɪ			
hameru	kaburu	suru	tsukeru
sport gloves	hats	watches*	watches*
sport masks		(finger) rings*	(finger) rings*
robbery masks		(thin) gloves*	(thin) gloves*
		(surgical) masks*	(surgical) masks*
		(pierced) rings	elbow pads
		glasses	knee pads
		scarves	bras
			belts
			necklaces
			bracelets
			anklets
			earrings
			hair accessories

Table 6: Data for Accessories

The results show that the Cantonese verb *taɪ* translates to four Japanese verbs: *hameru, kaburu, suru,* and *tsukeru. Hameru* is used for garments made of thick material only, such as baseball gloves and face masks for hockey goalies. *Kaburu* is used for non-accessory garments worn on the head, such as hats. *Suru* is basically a slang term used for idioms so it shares many items with *tsukeru.* It translates more closely as 'having' something than 'wearing' something. Examples of wearable garments for *suru* include items

[16] These sentences, along with their Japanese and Cantonese translations are provided in appendix A.

[17] For a description of IPA symbols, the reader is referred to Pullman & Ladusaw [25].

ta	t͡sʰa	t͡sœk		
tsukeru	nuru	haru		kiru
ties	make-up lipstick	underwear shoes socks nylons rollerblades boots	(shorts)	sweaters shirts kimonos overalls jackets undershirts suits tuxedos

Table 7: Data for Ties, Make-up, and Clothing

that can "hang" onto a person, such as glasses, scarves, or pierced rings. Other items include watches, finger rings, thin gloves, and thin masks that do not encompass the whole head. Accessories[18] such as jewelery and protective gear require the verb *tsukeru*.

From table 7, we see that in Cantonese, the verb *ta* is used for ties while Japanese uses the verb *tsukeru*. This suggests that ties are either treated as accessories (as in the case of Japanese), or they are treated as a separate category (as in Cantonese). A more cross-linguistic comparison is needed before the classfication for ties can be made.

The rest of table 7 deals with wearable garments that are considered as make-up and clothing items. For make-up, both Cantonese and Japanese use one verb, which are *t͡sʰa* and *nuru* respectively. The meaning for both *t͡sʰa* and *nuru* is "to brush something onto skin". These verbs describe the motion associated with make-up while the English verb *to wear* simply describes the state of the event. Lastly, the Cantonese verb *t͡sœk* is used with clothing item such as shirts, jackets, suits, shoes, and socks, whereas Japanese uses *haku* for clothing items that cover anything below the waist only and *kiru* for items that cover the upper body or the entire body.

Based on the data collected from English, Cantonese, and Japanese, we can hypothesize a word hierarchy for the verb 'wear' shown in figure 7.

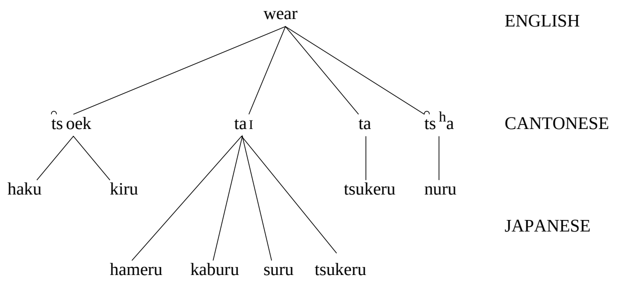

Figure 7: The 'wear' Hierarchy

Figure 7 shows that the English verb 'wear' corresponds to four different words in Cantonese, namely 't͡sœk', 'taɪ', 'ta', and 't͡sʰa'. These Cantonese verbs in turn correspond to multiple words in Japanese. The verb 't͡sœk' corresponds to 'haku' and 'kiru', the verb 'taɪ' corresponds to 'hameru', 'kaburu', 'suru', and 'tsukeru', the verb 'ta' corresponds to 'tsukeru', and the verb 't͡sʰa' corresponds to 'nuru'.

How can we use this information to help MT? The author restructured the hierarchy into one which an MT system can make use of. Figure 8 shows the hierarchy we can use for translation purposes.

[18] Note that belts and bras are classified with other types of accessories but not other types of clothing items.

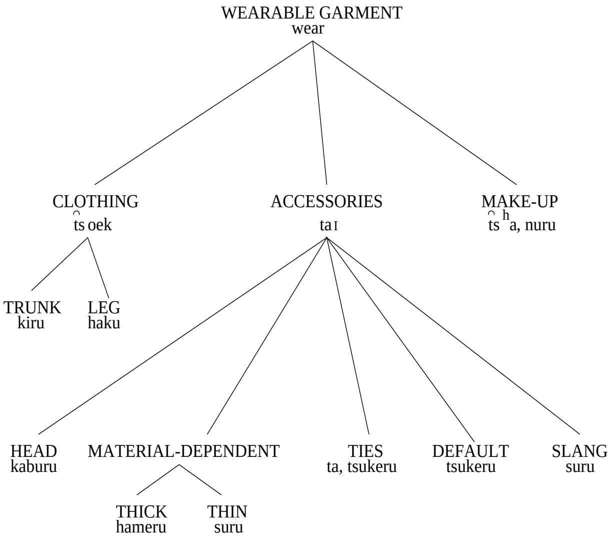

Figure 8: The Translation Hierarchy

What we have done is basically superimposed a word hierarchy for ‘wear’ from three languages into one, and called it “the translation hierarchy”. Given this translation hierarchy, we can use it to determine which verb we use for different wearable garments cross-linguistically. For implementation purposes, figure 8 would need to incorporate a few more details. Each word in the tree needs to show which language it is part of (e.g., “E” for English, “C” for Cantonese, and “J” for Japanese). Another detail needed is monolingual information. All the words in one language which are leaf nodes need to be indicated as such. In particular, in English, ‘wear’ is a leaf node because none of its children nodes are English words. In Cantonese, ‘t͡sœk’, ‘t͡sʰa’, and ‘ta’ are leaf nodes. As for Japanese, all the words are leaf nodes. Finally, each wearable garment needs to be classified into one of the groups used in the tree (so a bracelet belongs to “Accessories”, shoes belong to “LEG”, etc.).

Now, suppose we were translating the sentence “Watashi wa jakketo wo kite imasu (I am wearing a jacket)” from Japanese to English. First, we start at the top of the hierarchy. The top node “Wearable Garment” has the general verb ‘to wear’. We see that it is a leaf node so without traversing the tree any further, we choose ‘wear’ as the appropriate verb. Suppose we were translating the sentence “I am wearing a tie” from English to Cantonese. Once again, we start off at the top node “Wearable Garment”. We need to find the category to which a tie belongs, which is “Ties”. We traverse the second level of the tree and we find “Accessories”. Since “Ties” fall under “Accessories”, we traverse the children of “Accessories” and find “Ties”. Then we see that the verb ‘ta’ is a leaf node, so we stop and determine that the equivalent of ‘wear’ for a tie in Cantonese is ‘ta’.

We mentioned earlier that such a hierarchy can make predictions for items that are not in the lexicon. How does this work? We observe that the translation for an anklet in Japanese is literally “a bracelet at the ankle”. The verb used for an anklet is the same one used for a bracelet, *tsukeru*, not the one used for wearable items on the leg. Therefore, we can predict that if toe rings become a new fashion, the translation for it in Japanese would be “a ring at the toe”. Even though a toe ring is worn on the foot, we would not use *haku*, but rather *tsukeru* (the same verb used for a finger ring) or *suru* (the same verb used for slangs) would be used instead.

4.2.5 Recognizing Compounds

As we mentioned earlier, identifying compounds can be quite difficult. Fortunately, there are several properties of compounds that we can exploit to help us recognize them as one unit. This section discusses the properties that are specific to English compounds.

Consider the sentence ‘Sally drop kicked a soccer ball’ with the compound ‘drop kick’. Notice that the past tense marker `-ed` is inflected on the entire compound ‘drop kick’ so we do not get *‘dropped kick’.

Similarly, consider the sentence 'The fox hunters are out to catch them again' with the compound 'fox hunter'. Notice that the plural marker `-s` is inflected on the entire compound so we do not get *'foxes hunter'. Identifying tense and plural markers can help recognize compound units when adjacent words are lexically ambiguous.

Another property is the presence of the degree word 'very'. A degree word modifies an adjective but not a noun. Knowing this helps us parse the phrase 'a very green house'. Since we know that 'very' modifies an adjective, the following word 'green' must be an adjective. Therefore, 'house' must be the head noun of the phrase, so 'green house' cannot be a compound.

A third property is exhibited in irregular plurals in English. Words such as 'leaf', 'tooth', 'foot', and 'man' have irregular plurals in most contexts: 'leaves', 'teeth', 'feet', and 'men' respectively. However, with compounds, these words do *not* take their expected irregular forms. Instead, they follow the regular `-s` plural. Examples that illustrate this phenomenon are (taken from O'Grady [21]): 'Maple Leafs', 'saber tooths', 'bigfoots', and 'walkmans'. Since the dictionary keeps track of a list of words that take an irregular plural form, then we know that we have a compound when these words undergo the regular plural instead.

4.3 Information in the Lexicon

If we are trying to simulate language production and interpretation (either because we are using the machine to complete a text translation task or to communicate with another human), then we need to understand how the human mind works and model it. From this point of view, we need to incorporate research from different fields together in order to simulate the human mind. These fields include artifical intelligence, linguistics, psychology, philosophy, neurobiology, and anthropology. However, two obstacles arise. First of all, neurobiology has made little advance in helping us learn how the human brain functions. Therefore, our models are limited to psychological approaches with insight from neurobiology.

Secondly, even though we may have sufficient results from various research areas, it is an overwhelming task to incorporate the vast knowledge into one system. When humans translate, we use at least five different kinds of knowledge available to us. These are (quoted from Arnold *et al.* [2]):

(1) Knowledge of the source language.

(2) Knowledge of the target language. This allows [human translators] to produce texts that are acceptable in the target language.

(3) Knowledge of various correspondences between source language and target language (at the simplest level, this is knowledge of how individual words can be translated).

(4) Knowledge of the subject matter, including ordinary general knowledge and 'common sense'. This, along with knowledge of the source language, allows them to understand what the text to be translated means.

(5) Knowledge of the culture, social conventions, customs, and expectations, etc. of the speakers of the source and target languages. [This knowledge] is what allows translators to act as genuine mediators, ensuring that the target text genuinely communicates the same sort of message, and has the same sort of impact on the reader, as the source text.

The first two kinds of knowledge involve a dictionary with entries from the source and target language as well as syntactic rules that generate grammatical sentences. The third kind of knowledge is mostly dealt with by the underlying model that an MT system uses. For example, a direct MT system has the simplest type of information that translates from SL to TL with word level knowledge only. The fourth kind of knowledge that Arnold *et al.* states is often referred to as *world knowledge*. This information is the sort of information that people can say "Everybody knows that!" to. It also includes *default* knowledge, such as knowing that the grass is usually green and the sky is usually blue. The last type of knowledge is cultural information available from anthropology. The reader is referred to Hatim & Mason [12] for translations that require cultural information.

Because the study of lexicon is a very wide topic, our discussion will emphasize the first three kinds of knowledge mentioned above. The rest of this section is divided into three parts. First, we examine typical lexical entries in a paper dictionary. Then we briefly survey the kind of information that is in a human lexicon. Lastly, we explore the kind of information that is needed in an MT dictionary.

4.3.1 A Paper Lexicon

This subsection will examine the kind of information that is available in a paper dictionary. Consider the following dictionary definition of the word 'glue' (taken from the Oxford Advanced Learners Dictionary [13]):

> **glue** /`glu:; glu`/ *n* [U] thick, sticky liquid used for joining things, eg broken wood, crockery. □ *vt* (*pt, pp* `glued`; *pres p* `gluing`) [VP6A, 15A, B] ~ **(to), 1** stick, make fast, with ~: ~ *two pieces of wood together;* ~ *a piece of wood on to something.* **2** put tightly or closely: *His eyes were/His ear was* ~*d to the keyhole. Why must you always remain* ~*d to your mother*, Why can you never be separated from your mother? **~y** /`'glu:i ; 'glui`/ *adj* sticky, like ~.

The entry shows that the word 'glue' is listed as the first item (called the *headword*) in the entry. There is only one syllable in 'glue' because letters in the headword are not separated by a dash. The second item is the phonetic pronunciation of the word. This entry shows that there are two acceptable ways of pronouncing 'glue', one with a long vowel (denoted by ':'), and the other with a regular vowel.

The first definition of 'glue' is an uncountable (shown as "[U]") type of noun that means "a sticky liquid used for joining things". Because the word 'glue' can be used as a noun or a verb, the square □ is shown to separate the two usages. The second definition is a verb that has irregular conjugation. The past tense of 'glue' is not the regular suffixation of `-ed` and the progressive tense is not the regular suffixation of `-ing`. Instead, 'glue' needs its final `e` removed before the suffixes `-ed` and `-ing` can be attached. The list [VP6A, 15A, B] indicates which verb pattern 'glue' belongs to. This information includes *syntactic subcategorization* and the *modality* of the verb. The entry shows that the verb 'glue' has two senses, which are enumerated with their definitions and examples. Finally, when the word 'glue' is suffixed with `-y`, it becomes an adjective. The corresponding pronunciation, lexical category, and definition follow.

Having examined a paper dictionary entry, we will go onto the next subsection that compares the differences between entries in a paper dictionary and entries in a human dictionary.

4.3.2 A Human Lexicon

Is it possible to know how the human mind is structured? Probably not, as it is impossible to know the internal workings of a biological being without dissecting it. However, we can gain insight on the human lexicon by finding clues from daily conversations (e.g., word searches and slips of the tongue), theoretical linguistics, speech disorders, and psycholinguistic experiments. This section presents some conclusions drawn by Aitchison [1] about the human lexicon. Example sentences in this section are taken from Aitchison [1] unless otherwise specified. The reader is referred to her book for an excellent overview of the mental lexicon from a psycholinguistic perspective.

The first question we ask is, what is a human lexicon? The intuitive answer is that a human lexicon is a part of the brain that stores information about words. However, it is hard to give a precise definition, because no one knows exactly what kind of information is stored in our minds. What we will do instead is to gather ideas from various researchers and come up with some properties of the human lexicon that most researchers agree upon.

Just how is the lexicon structured? Is it the same as a paper dictionary? What kind of information do we store? Does everyone store the same information?

We know that our lexicons are structured in an orderly fashion. There are two strong pieces of evidence that support this claim. One is due to the fact that humans know so many words. Seashore & Eckerson (1940) and Diller (1978) (from Aitchison [1]) conducted experiments which showed that an average college student knows 150,000 to 250,000 words[19]. The second reason is due to fast retrieval time for a word in regular speech. Lennerberg (1967) and Marslen-Wilson & Tyler (1980, 1981) (from Aitchison [1]) showed that on average, native speakers produce six syllables per second and recognize a word one-fifth of a second after hearing the *onset*[20] of a word in their language.

Although we know that our lexicons are well structured, it is hard to imagine that the entries in our lexicons are stored alphabetically. When we make speech errors, we often replace the word we intend

[19] A *word* in their experiments included derivations of the same root as well as distinct roots. For example, 'loyal' and 'loyalty' were considered as two words in their experiments.

[20] An onset is the beginning of a syllable. In languages with vowels, a syllable can be broken into three parts: onset, nucleus (vowel), and coda. The consonants before the nucleus constitutes the onset.

to say with a word that has a similar meaning. For example, “The inhabitants of the car were unhurt” should have the word *occupants* instead of *inhabitants*. This kind of speech error suggests that our lexicons are ordered by meaning rather than spelling. However, some speech errors suggest that at least part of our lexicons are orderly by affixes. For example, “The doctor listened to her chest with his *periscope* (stethoscope)” suggests that the speaker retrieved a word with the common suffix, `-scope`. Another example is “He told a funny *antidote* (anecdote)” where the speaker retrieved a word that has common prefixes. Perhaps then, our lexicons are mainly organized by meaning, and partially organized by affixes.

We saw in the previous section that a paper dictionary entry gives grammatical information and a concise definition for each sense of the word. Several differences arise from this description. One major difference is that entries in our minds are likely to contain more information than just syntactic information and context-free definitions. If we were asked for the definition of a ‘photograph’, we may visualize an image of a photograph before we come up with the meaning. The associated image of the photograph may be a picture of a birthday party or a painting from a museum. We may also remember the events that happened relating to that photo, such as a joke that somebody told at the party or the artist of the painting. This example shows that words are linked to other information in our minds. Definitions can also be based on personal experiences. For example, if we were asked to describe a car, the amount of detail in the description depends greatly on how much we know about cars or about the interrogater. If we were asked to explain what ‘moral values’ are, we would most likely draw on examples from past experiences to illustrate what moral values mean to us. Since experience differs from person to person, we often have different definitions for the same words. If a foreign speaker pointed to an object on a table and asked us whether it is a vase or a bowl, we would say “a vase” if it is long and thin and we would say “a bowl” if it is low and flat. However, if the object is the shape of a tall glass, we may decide that it resembles our image of a vase more closely. This example shows that we have *default* or *generic* definitions and we often use features to rank the resemblence of an object. Finally, if somebody asked us “what is a mug?”, we would reply by asking for the context of the ‘mug’. If it is an action related to theives and victims, then we say that the meaning is a robbery. On the other hand, if it is an object related to kitchenware, then we say that the meaning is a cup. Unlike a paper dictionary, we do not just list out all the possible meanings for an isolated word, we request for clarification by providing the word in context.

Although no dictionary can ever be complete, a paper dictionary is static whereas a human dictionary is dynamic. “Not only do [paper] dictionaries generally restrict themselves to either general, or specialist technical vocabulary (but not both), in addition, new words are constantly being coined, borrowed, used in new senses, and formed by normal morphological processes. (Arnold *et al.* [2])” It is important for us to realize that languages are constantly changing, because updating the lexicon and storing large amounts of information are the main advantages that machines have over humans and books. These advantages will serve as the basis of our discussion in the next section.

4.3.3 A Machine Lexicon

In an MT system, the lexicon is the most important component because “the size and quality of the dictionary limits the scope and coverage of a system, and the quality of translation that can be expected. (Arnold *et al.* [2])” For systems that are not domain specific, building a machine lexicon can be very overwhelming. Compiling a machine lexicon is more complicated than compiling a paper dictionary because a machine needs (almost) all the information in a paper dictionary but for multiple languages. Furthermore, some machine lexicons have association links and hierarchical structures, which means that more work is required for better design.

Due to different theories and implementation styles, existing MT dictionaries are “diverse in terms of format, coverage, level of detail and precise formalism for lexical description. (Arnold *et al.* [2])” However, MT dictionaries have two components in common: (i) the monolingual module and (ii) the translational module.

Each entry in the lexicon must store monolingual information such as syntactic subcategorization and semantic restrictions. Syntactic subcategorization specifies the kinds of phrases that can “follow” other phrases. For example, the sentence “Tom gave a book to Jeremy” is grammatically correct because the verb “to give” takes two *complements* (or must be “followed” by two phrases): a direct object and a recipient of the object. More specifically, the direct object must be a noun phrase (NP) and the recipient can be a preposition phrase (PP), as in our example, or a noun phrase, as in “Tom gave Jeremy a book”.

This formalism is defined by a syntactic theory called *Argument Structure* (Grimshaw [10]).

Verbs are not the only type of lexical entries that have subcategorization restrictions. Some examples of subcategorizations include the following: nouns can take PP's as complements, as in "the destruction *of the city*", just as prepositions can take NP's as complements as in "at *the corner store*"; and adjectives can take PP's as complements, as in "proud *of herself*". The reader should note that subcategorization only specifies syntactic restrictions; Argument Structure alone will allow nonsensical sentences such as "Tom gave the storm to his pen". This sentence satisfies all the syntactic properties, such as having 'give' take an NP and a PP. However, the sentence does not make sense, which means that we have violated semantic restrictions.

Semantic restrictions classify words into different groups, such as human versus non-human, animate versus inanimate, and abstract versus concrete. For example, we know that the verb 'to melt' takes an NP complement, but the NP must be an object that is *meltable.* We also know that the verb 'to scream' requires that the agent of the action to be animate, otherwise it is unable to utter the sounds. Both semantic restrictions and syntactic subcategorization must be present in the lexicon to produce meaningful translations.

The second component in an MT dictionary is the translational module. This module consists of translation rules that relate words in SL to words in TL. This paper has provided plenty of evidence to show that translation rules cannot be as straightforward as mapping a word in SL to its equivalent in TL. Translation rules are specified based on the underlying theory of the system. For example, one system may use *inheritance*, another may not. A discussion on the various approaches taken are beyond the scope of this paper, but we will briefly explain the usage of inheritance in MT.

Since machine dictionaries are complicated and much information is redundant, we need to reduce this redundancy in some way. Linguistic theories can provide theoretical basis for lexical rules so that we can eliminate *horizontal redundancy.* For example, rather than storing 'car' and 'cars', we only store the singular form and apply a lexical rule to produce (or reduce) the plural form when necessary. Many systems employ multiple inheritance to eliminate *vertical redundancy.* For example, the entry for a 'puppy' may be a child node of the entry for a 'dog', which in turn may be a child node of an 'animal'. These relationships allow the individual entries to inherit semantic information from their parent nodes and they help the system perform inferences. Inheritance is a well studied topic in lexicography, so we will not discuss any details here.

4.4 Lexical Choice

One of the central issues in MT is lexical choice. As mentioned earlier, lexical ambiguity exists when a word in one language has more than one sense in another language. Therefore, the accuracy of the translation depends heavily on the correct choice of word sense. This section presents this problem from a natural language generation (NLG) perspective rather than from a pure MT perspective for two reasons. First of all, problems in lexical choice exist for both research areas. Secondly, machine translation has not taken advantage of the various approaches by NLG researchers to this problem. Since lexical choice is a huge problem in general, this section reviews the main issues involved in lexical choice without providing solutions to the problem. For various approaches in tackling this problem in natural language generation, the reader is referred to McKeown [17].

There are four major ways in which lexical choice can affect NLG:

(1) syntactic subcategorization,

(2) the amount of information to be communicated,

(3) what additional information can be added, and

(4) relative salience of the elements being expressed.

Subcategorization[21] can affect lexical choice in three ways. The first is the quantity of subcategorization. For example the sentence "John arrived" is grammatical but "*John arrived [the pen]"[22] is not. Therefore, we cannot just randomly select phrases and concatenate them together to form a sentence.

[21] The definition of subcategorization was provided in section 4.3.3.

[22] In this section, the boundary of a syntactic phrase is marked by square brackets.

Secondly, the type of subcategorization is important. For example, "The boy thinks [that skiing will be fun]" is grammatical but "*The boy thinks [the ice]" is nonsense because a piece of ice is not a kind of thought. In particular, the verb 'to think' takes *complementizer phrases* (CPs)[23] as complements and not NPs. The third is the meaning of subcategorization. For example, "The boy melted the ice" is expressible but not "*[The ice] melted [the boy]". In this example, we cannot simply use syntactic knowledge to specify the subcategorization (i.e. choosing CPs over NPs), but we must know the semantics behind different types of NPs (i.e. the NP [the ice] is "meltable" but the NP [the boy] is not). All of these issues are dealt with in the theory of Argument Structure.

The second problem in lexical choice is the amount of information to be communicated. Consider the difference between "Floyd **arrived** safely in Boston" and "Floyd **landed** safely in Boston". Both sentences[24], are grammatical sentences and syntactically identical. However, we do not know how Floyd got to Boston in the former sentence while we can infer that Floyd went to Boston by means of an airplane in the latter. This difference is subtle but relevant. A related example is "Sally went to Boston". To a human, we automatically infer that Sally intended to go to Boston and she arrived there, but we do not express it explicitly. To a machine, it may be necessary to change the sentence to "Sally went to Boston *and got there*".

The third problem is what other information can be added. This problem is not simply choosing the correct syntactic phrase, but it involves knowing whether the sentence is talking about an on-going action or a completed action. For example, "Peter was deciding [for an hour]" is grammatical because the PP [for an hour] is a possible complement for the progressive form of the verb "to decide". However, both "*Peter decided [for an hour]" and "*Peter made a decision [for an hour]" are ungrammatical because they are completed actions.

Relative salience is the last problem in lexical choice that we will discuss in this section. A simple example that illustrates the problem is found in the following sentences: "The car in the garage is **green**" and "The green car is **in the garage**". Both sentences are talking about a green car being in a garage, but they have different emphasis on what is being stated. Much research has been devoted to studying the role of salience in NLG for the past two decades. Some aspects of NLG which salience affects are *content selection*, *content ordering*, *lexical selection*, and *reference generation.* Since the salience of the elements being expressed relates to how humans perceive and understand natural language, psychology can play a big role as well. For a psychological account on salience and its effects in NLG, the reader is referred to Pattabhiraman's thesis [22].

4.5 Summary

This section gave a quick overview of stemming, lexical ambiguity, the structure and information stored in three kinds of lexicon, and lexical choice. We saw examples of how these problems affect the accuracy of translations and in some cases, we studied some approaches to solving these problems. In general, these problems have been around since the beginning of natural language processing and no single solution is good in both practice and theory.

The subsection on lexical choice looked at four major ways in which the accuracy of NLG can be affected. Although the psycholinguistic aspects of lexical choice were not the topic of this section, it is important that we understand how humans search and select the words they say and write. Aitchison [1] surveys several types of tongue slips, including *blending*, *blocking*, and *Freudian slips.* She provides an analysis of three models to account for the human word selection process. An important point she makes is that in language production, "it is normal for the mind to activate many more words than are likely to be used in the course of a conversation". This point gives us some insight on how a machine should generate sentences.

[23] A complementizer phrase usually has a "connective" word as the head of the phrase. These words can be 'that', 'because', 'so', 'who', and so on.

[24] For the rest of this subsection, example sentences used to explain how much information is to be communicated, what additional information can be added, and relative salience are taken from Meteer [18].

5 Comparing Different Techniques

To obtain a better understanding of the difficulty in stemming English affixes, the author has devised a program that compares different stemming techniques. This program, *stripper*, is described in section 5.1. Following these stemming algorithms, the reverse process of stemming is also surveyed for comparison purposes. This program, *affixer*, is described in section 5.2. Neither of these programs have been implemented fully, so only the design of the programs will be described.

5.1 The Stripper

The stripper removes affixes on a word-by-word basis. Given a word, the program will remove affixes if it follows the specified algorithm. There are three versions of the stripper. The first version employs a pure pattern matching technique. The second version recycles the first version but with access to a dictionary. Finally, the third version is an implementation of the Porter algorithm, which we discussed earlier in section 4.1. Below is a more detailed description of these versions.

Version 1 of the stripper takes a word and an affix as input and concatenates the two to generate the output. For example, given 'singing' and `-ing`, the program will output 'sing', and given 'unhappy' and `un-`, the program will output 'happy'. However, it also generates '*sl' given 'sling' and `-ing`. Version 2 tackles this problem by using a dictionary with some heuristics before generating the output. The heuristics used are:

- if the non-processed word is already in the dictionary, then do not process it
- if the processed word is not in the dictionary, then find its "nearest neighbour" in the dictionary
- if the processed word does not contain any vowels, then do not remove the suffix

These heuristics have not been conceptualized fully yet. One problem is that the term "nearest neighbour" has not been defined. The intuition behind finding a nearest neighbour is to linearly look up words in the dictionary with the processed word. However, should we look up every entry in the dictionary that *contains* the processed word? That would entail matching a substring for every lexical entry in the dictionary, which seems computationally unreasonable. Suppose we found a small list of such entries successfully. How do we determine which one is the *closest* neighbour? Another problem is whether these heuristics should be used together or separately. The author has left these questions open.

Version 3 implements the Porter stemmer [24]. There are five steps in the algorithm, which are listed in appendix B.

5.2 The Affixer

The affixer concatenates affixes on a word-by-word basis. Given a word and an affix, the program will attach the affix onto the word if it follows the algorithm used. There are three versions of the affixer. Like the stripper, the first version employs a pure pattern matching technique. The second version recycles the first version with the addition of some morphological knowledge we discussed in this paper. The last version is an implementation of the algorithm used by Dorr [6]. We describe these versions in more detail below.

Version 1 of the affixer takes a word and an affix as input, and generates a new word as output. For example, given 'sing' and `-ing`, the output will be 'singing', and given `un-` and 'happy', the output will be 'unhappy'. However, it also generates '*happying' given 'happy' and `-ing`. To solve this problem, version 2 uses a dictionary and determine that 'happy' is an adjective that does not take an `-ing` suffix. So the output will be an error message instead. Version 3 tries to solve the problems found in version 2 by using rules from Dorr's algorithm. For the rest of this section, we will go over what these rules are without evaluating on its effectiveness.

The notation used to illustrate Dorr's rules are:

- V is a vowel
- C is a consonant

- C_1 is one of: b, d, f, g, l, m, n, p, r, s, t
- C_2 is any consonant except: c, g
- Q is any letter except: i, a
- S is one of: ch, sh, s, z, x
- G is one of: g, c
- E is one of: e, i
- ∅ is null

Dorr makes use of five rules in her system. They are epenthesis, gemination, y-replacement, elision, and i-replacement. The rules shown here are written in the form of `A → B / C _ D`, which means that `A` changes to `B` if A is preceded by `C` and followed by `D`.

Epenthesis:	s → es / S _
Gemination:	$C_1 \rightarrow C_1C_1$ / CV _ V
Y-Replacement:	y → i / C _ Q
Elision:	e → ∅ / C_2 _ V
	e → ∅ / CV _ e
	e → ∅ / G _ E
I-Replacement:	ie → y / _ i

Epenthesis states that a plural `-s` becomes `-es` if it appears after a letter of class S. Gemination states that a consonant of the class C_1 geminates if it follows a consonant and a vowel and precedes another vowel. Y-replacement states that a 'y' ending word becomes an 'i' if the 'y' follows a consonant and precedes either an 'i' or an 'a'. Elision states that an 'e' is deleted if it appears in one of the three contexts: (i) after a consonant of class C_2 and before a vowel, (ii) after a consonant and a vowel and before another 'e', and (iii) after a consonant of class G and before a vowel of class E. Lastly, i-replacement states that 'ie' changes to a 'y' if it appears before another 'i'.

6 Conclusions and Future Work

> Many people originally hoped that computers would slice through human babble to usher in a new era of understanding. Almost as soon as computers became a reality, people wanted to use them as translators, thinking the equation WORDS + GRAMMAR = LANGUAGE would do the trick. But a machine that doesn't process a language's spirit can only turn out a corpse of speech. Analogy, innuendo, context, nuance, cultural reference — a creation that can't grasp these and other shadings of speech will never be able to make sense of human conversation. (Jennings [16])

This quotation summarizes the history of machine translation and the attitudes that researchers have had in the past. Although progress in MT has been slow, advances have been accomplished and many commercial systems are available. Researchers are now solving many problems that were overlooked before.

We studied a specific aspect of MT — the contribution of morphology. Without any empirical evidence, it is impossible to conclude just how important the role of morphology is to MT. Nonetheless, we studied various morphologically related problems that are associated with the different theories of MT. We started by surveying three approaches that illustrate the wide spectrum of linguistic relevance in MT. These approaches are direct, transfer, and interlingua. Although each of these approaches have advantages and disadvantages in terms of theory and practice, we concluded that none of them were ideal.

Following the survey was an introduction to morphological theory. We defined various terms and studied examples for derivation, inflection, compounding, clipping, and blending. To sufficiently account for numerous morphological phenomena, we claimed that a two-level morphological representation is needed in MT.

With this morphological background, we tackled various morphologically related problems in MT. Firstly, we looked at the applicability of stemming in MT. Then we looked at how different types of lexical ambiguities (e.g., categorial ambiguity, homographs, homophones, polysemes, and translational ambiguity) are handled with the aid of morphological knowledge. We also sampled data from Japanese and Cantonese to illustrate the idea of word hierarchies — a new proposal to help solve the problem of translational ambiguity. Since the analysis was limited to two languages and data collected from two speakers, the results are preliminary and need to be carried out further before a sound theory can be integrated into an MT system. Thirdly, we compared three kinds of lexicons: paper lexicon, human lexicon, and machine lexicon. This comparison allowed us to appreciate the degree of difficulty in creating a machine lexicon for translation. Lastly, we briefly reviewed the major problems in lexical choice from a general view of natural language processing.

The last part of this paper proposed some heuristics and compared different techniques used for morphological analyzers in MT. In particular, we illustrated the functionality of two processes. One process is called "the stripper" which takes words and *strips* them down to their roots. The other process is called "the affixer", which takes words and appends affixes onto them. Both of these processes are described based on an English data set and neither of them have been fully implemented. The evaluation of these processes are left as future work.

Possible directions for future work include integrating findings from linguistic universals and typologies to help structure some ideas in MT. Areas of linguistics that may serve this purpose include universal word order, derivational categories, personal pronouns, number systems, body-part terminology, and noun classes. For a linguistic discussion on these topics, the reader is referred to Greenberg [9] and Talmy [28].

Another avenue is to investigate the effectiveness of a thesaurus in MT. The stages where the system builds an internal representation of the input and translates the representation into output may be able to take advantage of having access to a thesaurus.

There are still many problems and open questions in MT that were not mentioned in this paper. We did not consider issues related to phonology (for speech translation), syntax, semantics, discourse, and pragmatics. We feel that these are general problems in natural language processing (NLP) and that solutions from MT or NLP will give insight to the other.

Acknowledgements

The author would like to thank Dr. Nick Cercone for his valuable time and advice and endless encouragement throughout this project. She is a student member of the Institute for Robotics and Intelligent Systems (IRIS) and wishes to acknowledge the support of the Networks of Centres of Excellence Program of the Government of Canada, the Natural Sciences and Engineering Research Council, and the participation of PRECARN Associates Inc.

References

[1] Jean Aitchison. *Words in the Mind: An Introduction to the Mental Lexicon, 2^{nd}ed.* Blackwell, 1994.

[2] Doug Arnold, Lorna Balkan, Siety Meijer, R. Lee Humphreys, and Louisa Sadler. *Machine Translation: An Introductory Guide.* http://clwww.essex.ac.uk/~doug/book/book.html.

[3] L. Bauer. *English Word-Formation.* Cambridge University Press, 1983.

[4] John L. Beaven. Shake-and-bake machine translation. *Proceedings of the 15th International Conference on Computational Linguistics*, 2:602-609, 1992.

[5] P. F. Brown, J. Cocke, S. A. Della Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. S. Roossin. A statistical approach to machine translation. *Computational Linguistics*, 16:79-85, 1990.

[6] Bonnie J. Dorr. *Machine Translation: A View from the Lexicon.* MIT Press, 1993.

[7] Kenneth Goodman. Special issues on knowledge based MT, parts I and II. *Machine Translation*, 4(1-2), 1989.

[8] Kenneth Goodman and Sergei Nirenburg. *The KBMT Project: A Case Study in Knowledge Based Machine Translation.* Morgan Kaufmann, San Mateo, California, 1991.

[9] Joseph H. Greenberg, Charles A. Ferguson, and Edith A. Moravcsik. *Universals of Human Language, Volume III: Word Structure.* Stanford University Press, 1978.

[10] J. Grimshaw. *Argument Structure.* MIT Press, Cambridge, MA, 1991.

[11] Donna Harman. How effective is suffixing? *Journal of the American Society for Information Science*, 42(1):7-15, 1991.

[12] Basil Hatim and Ian Mason. *Discourse and the Translator.* Longman, London, 1990.

[13] A. S. Hornby. *Oxford Advanced Learner's English-Chinese Dictionary.* Oxford University Press, 1984.

[14] Bowen Hui. Analysis on the affix *-ism*. *Paper for LING 405 at the University of British Columbia*, 1995.

[15] W. John Hutchins and Harold L. Somers. *An Introduction to Machine Translation.* Academic Press, 1992.

[16] Karla Jennings. *The Devouring Fungus: Tales of the Computer Age.* W. W. Norton & Company, Ltd., 1990.

[17] Kathleen R. McKeown. *Text Generation: Using Discourse Strategies and Focus Constraints to Generate Natural Language Text.* Cambridge University Press, 1985.

[18] Marie Meteer. Bridging the generation gap between text planning and linguistic realization. *Computational Intelligence*, 7(4):296-304, 1991.

[19] M. Nagao. *A framework of a mechanical translation between Japanese and English by analogy principle.* In A. Elithorn and R. Banerji, editors, Artificial and Human Intelligence, p.173-180, North Holland, Amsterdam, 1984.

[20] Kanlaya Naruedomkul and Nick Cercone. Steps toward accurate machine translation. *7th International Conference Theoretical and Methological Issues in Machine Translation*, TMI '97:63-75, 1997.

[21] William O'Grady and Micheal Dobrovolsky. *Contemporary Linguistic Analysis: An Introduction,* 3^{rd}*ed.* Copp Clark Ltd, 1996.

[22] T. Pattabhiraman. *Aspects of Salience in Natural Language Generation.* PhD thesis, Simon Fraser University, Vancouver, B.C., August 1992.

[23] M. Popovic and P. Willet. The effectiveness of stemming for natural-language access to Slovene textual data. *Journal of the American Society for Information Science*, 43(5):384-390, 1992.

[24] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130-137, 1980.

[25] Geoffrey K. Pullman and William A. Ladusaw. *Phonetic Symbol Guide.* The University of Chicago Press, 1986.

[26] Harold L. Somers. *Current research in Machine Translation.* In John Newton, editor, Computers in Translation: A Practical Appraisal, p.189-207, Routledge, London, 1992.

[27] Andrew Spencer. *Morphological Theory.* Blackwell, 1991.

[28] Leonard Talmy. Language typology and syntactic description. *Lexicalization patterns: semantic structure in lexical forms*, Volume III, Grammatical categories and the lexicon, 1985.

Appendix A: List of Sentences for 'wear'

This appendix lists twenty-nine sentences English sentences. For each English sentence, there is either one or two corresponding Japanese sentence and one corresponding Cantonese sentence. The Japanese sentences are written in Romanji, while the Cantonese sentences are transcribed in IPA (International Phonetic Alphabet)[25] but without tonal markings. The Japanese and Cantonese verbs that correspond to 'wear' are typed in boldface. The list of sentences follow.

(1) Mary is wearing a sweater.
Mary wa setā wo **kite** imasu.
Mary **t͡sœk** kʌn laŋ sam.

(2) Steve is wearing a jacket.
Steve wa jakketo wo **kite** imasu.
Steve **t͡sœk** kʌn lau.

(3) Jane is wearing overalls.
Jane wa ōbāōru wo **kite** imasu.
Jane **t͡sœk** kʌn kɔn jʌn fu.

(4) Tom is wearing a blue suit.
Tom wa aoi sūtsu wo **kite** imasu.
Tom **t͡sœk** kʌn ŋam seɪk sʌɪ t͡sɔn.

(5) Jane is wearing red socks.
Jane wa akai kutsushita wo **haite** imasu.
Jane **t͡sœk** kʌn hɔn seɪk mʌt.

(6) Fred is wearing pink underwear.
Fred wa pinkuno pantsuwo **haite** imasu.
Fre **t͡sœk** kʌn fʌn hɔn seɪk tʌɪ fu.

(7) Jenny is wearing nylons.
Jenny wa taitsu wo **haite** imasu.
Jenny **t͡sœk** kʌn leɪ lɔn si mʌt.

(8) I am wearing only one shoe.
Watashi wa kutsu wo katahō dake ni **haite** imasu.
ŋɔ t͡siŋ hʌɪ **t͡sœk** kʌn jʌt t͡sɛt haɪ.

(9) John is wearing rollerblades.
John wa rolāburēdo wo **haite** imasu.
John **t͡sœk** kʌn jʌt tœɪ rolaplet.

(10) Tom is wearing a gold watch.
Tom wa kinno udedokei wo **tsukete** imasu.
Tom wa kinno udedokei wo **shite** imasu.
Tom **taɪ** kʌn jʌt t͡sɛt kʌm piu.

(11) Fred is wearing five rings.
Fred wa yubiwa wo itsutsu **tsukete** imasu.
Fred wa yubiwa wo itsutsu **shite** imasu.
Fred **taɪ** kʌn m t͡sɛt kaɪ t͡si.

(12) Mary is wearing gloves.
Mary wa tebukuro wo **tsukete** imasu.
Mary wa tebukuro wo **shite** imasu.
Mary **taɪ** kʌn sau tʰou.

[25] Please see footnote 17 for a reference to IPA.

(13) Jane is wearing a purple scarf.
Jane wa murasakino sukāfu wo **shite** imasu.
Jane **taɪ** kʌn jʌt tʰiu t͡si seɪk kɛn kʌn.

(14) Mary doesn't wear glasses.
Mary wa megane wo **shite** imasen.
Mary mo **taɪ** an kɛn.

(15) Jane is wearing a belly button ring.
Jane wa oheso ni piasu wo **shite** imasu.
Jane **taɪ** kʌn jʌt ko tʰou wan.

(16) I am wearing black hair clips.
Watashi wa kuroi hea kuripu wo **tsukete** imasu.
ŋɔ **taɪ** kʌn hak seɪk fat kip.

(17) Steve is wearing kneepads.
Steve wa hiza ate wo **tsukete** imasu.
Steve **taɪ** kʌn kœt wu t͡sau.

(18) Mary is wearing an anklet.
Mary wa buresurreto wo ashikubini **tsukete** imasu.
Mary **taɪ** kʌn kœt lin.

(19) I am wearing a hair band.
Watashi wa heabando wo **tsukete** imasu.
ŋɔ **taɪ** kʌn tʰau kʷu.

(20) Mary is wearing earrings.
Mary wa iya ringu wo **tsukete** imasu.
Mary **taɪ** kʌn ji wan.

(21) Jenny is wearing a white bra.
Jenny wa shiroi burajā wo **tsukete** imasu.
Jenny **taɪ** kʌn pak seɪk hɔn wʌɪ.

(22) Tom is wearing a silver necklace.
Tom wa ginno nekkuresu wo **tsukete** imasu.
Tom **taɪ** kʌn jʌt tʰiu ʌn seɪk kɛn lin.

(23) Jenny is wearing a silver bracelet.
Jenny wa ginno buresurreto wo **tsukete** imasu.
Jenny **taɪ** kʌn jʌt tʰiu ʌn seɪk sau lin.

(24) John is not wearing a belt today.
Kyō John wa beruto wo **tsukete** imasu.
John mo **taɪ** pʰeɪ taɪ.

(25) Jane is wearing elbow pads.
Jane wa erubōpaddo wo **tsukete** imasu.
Jane **taɪ** kʌn sau wu t͡sau.

(26) I am wearing baseball gloves.
Watashi wa yakyūno gurōbu wo **hamete** imasu.
ŋɔ **taɪ** kʌn kʷan kʰau sau tʰou.

(27) John is wearing a hat.
John wa bōshi wo **kabutte** imasu.
John **taɪ** kʌn jʌt tɛn mo.

(28) Jane is wearing lipstick.
Jane wa kuchibeni wo **tsukete** imasu.
Jane wa kuchibeni wo **nutte** imasu.
Jane **t͡sʰa** t͡sɔ hau t͡sʌn ko.

(29) Mary is wearing a tie today.
Kyō Mary wa nekutai wo **tsukete** imasu.
Mary kʌm jʌt **ta** tʰaɪ.

Appendix B: Description of the Porter Stemmer

To explain the Porter algorithm, we use the following notation, rules, and examples taken from Porter [24].

(1) *Consonant* is a letter other than A, E, I, O, U, or Y when preceded by a consonant

(2) *Vowel* is any letter that is not a consonant

(3) c is a consonant

(4) C represents a sequence of consonants of length greater than 0

(5) v is a vowel

(6) V represents a sequence of vowels of length greater than 0

(7) All words can be written in the form $[C](VC)^m[V]$. Where the contents of the square brackets are optional. m is called the *measure* of the word or word part. When m = 0, the null word case is covered.

Measure	Examples
m = 0	TR, EE, TREE, Y, BY
m = 1	TROUBLE, OATS, TREES, IVY
m = 2	TROUBLES, PRIVATE, ORRERY

Table 8: Examples of the *measure* of a word

(8) Rules for suffix removal are given in the form (condition) S1 → S2. This format states that if a word ends with the suffix S1 and the stem preceding S1 satisfies the given condition, then S1 is replaced by S2.

(9) *S indicates that the stem ends with the letter S

(10) *v* indicates that the stem contains a vowel

(11) *d indicates that the stem ends in a double consonant

(12) *o indicates that the stem ends with a cvc cluster, where the second c must not be w, x, or y

Furthermore, for each set of the rules, only the one with the longest matching S1 for the given word is applied.

Step 1a

Conditions	S1	S2	Examples
NULL	SSES	SS	caresses → caress
NULL	IES	I	ponies → poni, ties → ti
NULL	SS	SS	caress → caress
NULL	S	NULL	cats → cat

Step 1b

Conditions	**S1**	**S2**	**Examples**
v	ED	NULL	plastered → plaster, bled → bled
v	ING	NULL	motoring → motor, sing → sing
$m > 0$	EED	EE	agreed → agree, feed → feed

Step 1b1 (to be performed if the second or third rule of Step 1b is successful)

Conditions	**S1**	**S2**	**Examples**
NULL	AT	ATE	conflat(ed) → conflate
NULL	BL	BLE	troubl(ing) → trouble
NULL	IZ	IZE	siz(ed) → size
*d and not (*L or *S or *Z)	NULL	single letter	hopp(ing) → hop, tann(ed) → tan, fall(ing) → fall, hiss(ing) → hiss, fizz(ed) → fizz
$m = 1$ and *o	NULL	E	fail(ing) → fail, fil(ing) → file

Step 1c

Conditions	**S1**	**S2**	**Examples**
v	Y	I	happy → happi, sky → sky

Step 2

Conditions	**S1**	**S2**	**Examples**
$m > 0$	ATIONAL	ATE	relational → relate
$m > 0$	TIONAL	TION	conditional → condition, rational → rational
$m > 0$	ENCI	ENCE	valenci → valence
$m > 0$	ANCI	ANCE	hesitanci → hesitance
$m > 0$	IZER	IZE	digitizer → digitize
$m > 0$	ABLI	ABLE	conformabli → conformable
$m > 0$	ALLI	AL	radicalli → radical
$m > 0$	ENTLI	ENT	differentli → different
$m > 0$	ELI	E	vileli → vile
$m > 0$	OUSLI	OUS	analogousli → analogous
$m > 0$	IZATION	IZE	vietnamization → vietnamize
$m > 0$	ATION	ATE	predication → predicate
$m > 0$	ATOR	ATE	operator → operate
$m > 0$	ALISM	AL	feudalism → feudal
$m > 0$	IVENESS	IVE	decisiveness → decisive
$m > 0$	FULNESS	FUL	hopefulness → hopeful
$m > 0$	OUSNESS	OUS	callousness → callous
$m > 0$	ALITI	AL	formaliti → formal
$m > 0$	IVITI	IVE	sensitiviti → sensitive
$m > 0$	BILITI	BLE	sensibiliti → sensible

Step 3

Conditions	**S1**	**S2**	**Examples**
$m > 0$	ICATE	IC	triplicate → triplic
$m > 0$	ATIVE	NULL	formative → form
$m > 0$	ALIZE	AL	formalize → formal
$m > 0$	ICITI	IC	electriciti → electric
$m > 0$	ICAL	IC	electrical → electric
$m > 0$	FUL	NULL	hopeful → hope
$m > 0$	NESS	NULL	goodness → good

Step 4

Conditions	S1	S2	Examples
$m > 1$	AL	NULL	revival → reviv
$m > 1$	ANCE	NULL	allowance → allow
$m > 1$	ENCE	NULL	inference → infer
$m > 1$	ER	NULL	airliner → airlin
$m > 1$	IC	NULL	gyroscopic → gyroscop
$m > 1$	ABLE	NULL	adjustable → adjust
$m > 1$	IBLE	NULL	defensible → defens
$m > 1$	ANT	NULL	irritant → irrit
$m > 1$	EMENT	NULL	replacement → replac
$m > 1$	MENT	NULL	adjustment → adjust
$m > 1$	ENT	NULL	dependent → depend
$m > 1$	(*S or *T)ION	NULL	adoption → adopt
$m > 1$	OU	NULL	homologou → homolog
$m > 1$	ISM	NULL	communism → commun
$m > 1$	ATE	NULL	activate → activ
$m > 1$	ITI	NULL	angularity → angular
$m > 1$	OUS	NULL	homologous → homolog
$m > 1$	IVE	NULL	effective → effect
$m > 1$	IZE	NULL	bowdlerize → bowdler

Step 5a

Conditions	S1	S2	Examples
$m > 1$	E	NULL	probate → probat, rate → rate
$m = 1$ and (not *oE)	NULL	NULL	cease → ceas

Step 5b

Conditions	S1	S2	Examples
$m > 1$ and *d and *L	NULL	single letter	controll → control, roll → roll