A Transfer Dictionary for Words and Bigrams

Jorge Kinoshita

Escola Politécnica da Universidade de São Paulo
Departamento de Computação e Sistemas Digitais (PCS) - Brazil
email: jkinoshi@pcs.usp.br URL: http://jk.pcs.usp.br

Abstract. We present a transfer dictionary model that explains how words and bigrams can be translated from one language to another in a concise way. Translation of bigrams can be used to handle the syntactic differences between languages. The dictionary is being used in a English-Portuguese translation machine project, however, the transfer dictionary model is generic enough to be applied to other projects as well.

1 Introduction

Bilingual dictionaries from language L1 to language L2, made to humans, declares to each word (ex: *"give"*):
- one or more tags (obligatory). Ex: verb.
- one or more translations to each tag (obligatory). Ex: *"dar"* (in Portuguese).
- translation depending on other words (optional). Ex: *"give-up"* - *"desistir"*. In this example, the translation is associated to a bigram.
- idiomatic expressions based on the word (optional).

We present a dictionary model to transfer systems that is similar because it declares the same kind of information dealing with words and bigrams, although we are not yet dealing with n-grams (ex: idiomatic expressions). The difference is that our declaration must follow a strict syntax that declares both morphological and syntactic aspects (ex: words position) of L2. This paper is an evolution of [1] that studies the union of a database of bigrams and a hierarchical structure of nouns in a machine translation project. If we add some extra notation to the database of bigrams then we get the transfer dictionary that is presented here. In this paper, bigrams are 2 words that has some syntactic relationship. Many syntactic differences in L2 can be handled by translating bigrams according to the dictionary.
Our transfer system is a bit different from others.
Generally, the machine translation based on the transfer approach operates in three steps:
1) **analysis**: maps sentence As into a syntactic tree Ast.
2) **transfer**: maps the structure Ast into another tree Bst, through the use of some transfer grammar.
3) **generation**: maps the tree Bst into the sentence Bs.
The transfer grammar is a set of rules heavily dependent on the kind of structure resulted from the analysis. For instance [2]:

$[gov : know, subj : \$1, obj : \$2[NP]] => [gov : conhecer, subj : \$1, obj : \$2]$.
This rule specifies that if the verb *"know"* has a subject and an object of type NP (noun phrase) then it must be translated as *"conhecer"* in Portuguese (*connaitre* in French) instead of *"saber"*. This rule, in particular, can not be applied to the sub-sentence : *"... in order to know that man"* because the right side of the rule does not match the tree due to the lack of subject (*"know"* is in infinitive). This rule should focus only on two words (a bigram): the verb *"know"* and its object (ex: *"man"*). We assume that most of the transfer rules can (and should) be written focusing only on bigrams because they define a good context and tend to be translated in the same way.
We are building an English to Portuguese transfer dictionary for a machine translation project. Our translation process is:
1) **Analysis**. Link the words of an English sentence in pairs (by some dependency grammar). It is possible to associate a relation-tag (kind of linkage) to each bigram. For instance in *"know-man"*, the relation-tag is +VB_NN that means *verb-object*.
2) **Transfer**. Translate the bigrams according to the dictionary and assemble them, building the translated sentence S with Portuguese words given in radical form.
3) **Generation**. Put the words of S in their normal form based on their radical form.

2 The Transfer Dictionary

In the dictionary, we declare translation of words and bigrams. Today, the treatment of idiomatic expressions is out of this scope.

2.1 Translation of a Single Word

The translation of one word is based on its tag. The set of tags in Portuguese is different from English. For instance, in Portuguese, nouns are classified according to its genre. We created a notation that translates not only the word but also its tag. We are adopting the following symbols:
W - word. Ex: *woman*
W' - word translation. Ex: *mulher*
W^T - word tag according to the TREEBANK notation. Ex: *NN*
W'^T - tag of word translation. Ex: *NNf*. The tag W'^T can be:
(1) the same as W^T.
(2) derived from the W^T by appending some information to it. Ex: NNf is composed of NN (substantive) and f (genre feminine).
(3) different from W^T. For instance, in some cases, *MD ("Modal")* becomes a verb in Portuguese.
The declaration of single words in the dictionary follows the syntax:
$W\ W^T : Wg'$

Example: $woman\ NN\ :\ mulher.f$
where:
$Wg' = (W' \mid W'.X \mid W' :: X)$
$W'.X$ means: "Append string W^T to X generating W'^T". Ex: $mulher^T$ = NN appended to f = NNf.
$W' :: X$ means: "set X to W'^T". Ex: *can* MD : *poder* :: VBP .
Generally, words and tags in the dictionary are in their radical form. The agreement in genre, number and person for Portuguese words is made by the transfer phase and is not explained by the rules in the dictionary.

2.2 Translation of Bigrams

The translation of bigrams deals with many aspects as it is shown in the examples:
a) *yellow - man : homem - amarelo*;
b) *give - up : desistir*;
c) *like - orange : gostar de - laranja.*
A bigram is generally translated as another bigram (ex: a). In Portuguese, adjectives generally follow the noun. The adjective *"amarelo/yellow"* had its position exchanged with its head.
In example (b), *"give-up"* is translated as one single word.
In example (c), the Portuguese verb *"gostar"* is not directly linked to its object and requires the preposition *"de"*.
We propose a set of rules to deal with these facts in bigram translation. These rules can be organized in a dictionary. There are 2 kinds of entrances to bigrams:
1) *a full bigram* (ex: *give - up*). The two words of the bigram are declared.
2) *a partial bigram* that is composed of one word and a relation-tag (ex: *know - +VB_NN*).
First, we are going to explain the rule syntax in order to translate two words and then, show how this same syntax can be generalized to translate partial bigrams.

Relation Tag Notation . The two words of a bigram are in a dependency relation. For instance: in the pair *concern - with* we can say that *"with"* is subordinated to *"concern"*. We can associate a relation-tag to each bigram. The relation-tag is derived from the two morphological tags of each word. For instance, the relation-tag of (concern, with) is +VB_PREP (verb and preposition). These two tags (ex: VB and PREP) are ordered according to the sequence of words in English sentences. The sign "+" before the first tag shows that the first word is the head, otherwise the sign "-" is used. For instance, the relation tag of (big, house) is -JJ_NN. The relation-tag does not follow exactly the TREEBANK notation [6]. For instance: -NN_VBP, -NN_VBD, etc. are mapped to -NN_ROOT in order to give the notion of *"subject"*.

2.3 Translation of Full Bigrams

The translation rule syntax of a full bigram is:

$H\ H^T\ S\ RT : H'\ S'$

For instance:

$concern\ VB\ with\ + VB\_PREP : preocupar\ com$

where :
H = head-word;
H^T = tag of H according to the TREEBANK notation;
S = subordinated-word;
RT = relation-tag;
H' = translation of head-word;
S' = translation of subordinated-word.
The pattern ($H\ H^T\ S\ RT : H'\ S'$) is incremented in order to specify the following operations:

a)**Order declaration**. When a bigram is translated, the order of the 2 words can be different in Portuguese. For instance: *"Ethernet protocol"* is translated as *"protocolo Ethernet"*. The order can be specified by adding the sign "+" to S' (otherwise, the English order is maintained): S'+ means that H' comes after S' and +S' means that H' comes before S'. For instance:

$protocol\ NN\ Ethernet\ - NN\_NN : protocolo\ + Ethernet$

In the transfer phase, the order declaration imposes a movement of not only the dependent-word but also of the entire phrase bellow it. For instance: *"a very fast car"* is translated as *"um carro muito rápido"* (a car very fast). The fact that S' *"rápido"* (fast) was moved to a position after H' *"carro"* (car) made the entire phrase *"muito rápido"* (very fast) be moved after *"carro"*.

b)**suppression of S' or H'**. There are cases that S' or H' is an empty string. This is declared by the "minus" sign in the place of the translated word (generally the S'). For instance: The case of translating *"there + to be"* as *"existir"* is easily handled by the rule:

$be\ VB\ there\ - EX\_VB : existir\ -$

c)**insertion of word**. A word in Portuguese can be inserted in the translation. For instance, *"like"* is generally translated as *"gostar"* plus the insertion of the preposition *"de"* . For instance, *"like chocolate"* is declared as:

$like\ VB\ VB\_NN-\ chocolate : gostar < -de\ chocolate$

The notation $H' < -P$ means that P must be placed before the phrase that has S' as its head.

2.4 Translation of Partial Bigrams

Instead of writing many bigrams (know, English), (know, John), etc. and their translations, it is possible to omit the object of *"know"*:

$know\ VB\ < VB\_NN- : conhecer$

The general statement is:

$H\ H^T < RT : H'$ or $H\ H^T < RT : H'\ S'$ (head in focus) and

$S\ S^T > RT : S'$ or $S\ S^T > RT : H'\ S'$ (subordinated-word in focus).
For instance, in Portuguese it is common to suppress the translation of *"it"* when behaving as the subject (-NN_ROOT) of some sub-sentence. This fact is expressed as:
$it\ PRP > -NN\_ROOT : -$

The *"no translation"* **Symbol - % :** It is possible to refer to some translated word W' (H' or S') without imposing any translation. This is helpful when we want to specify some attribute of W' such as its position in the translated sentence or its morphological tag. When the "%" symbol replaces W' in a statement S, then S does not impose any translation to W. Other rules will take care of translating W. For instance: the rule ($H\ H^T < RT : H'\ S'$):
$boy\ NN < -JJ\_NN : menino\ +\%$
means: any word referred by "%" (JJ/adjective) must come after *"menino"*. It can be applied to $(yellow, boy) => (menino, amarelo)$.

2.5 Relation-Tag-Only Based Translation

Other transfer rules that does not depend on words can be created guided by the above pattern. These rules are not declared in the transfer dictionary, but are used in the transfer phase. We decided to show these rules because they were derived from the same syntax of the dictionary. The rule syntax is:
$S^T << RT : H'\ S'$
or:
$H^T >> RT : H'\ S'$
In these rules, H' and S' are generally expressed by % . For instance:
$NNP >> -NN\_NN : \%\ +\%$
Ex: *"Ethernet protocol"* becomes *"protocolo Ethernet"*.

3 Using the Dictionary in the Transfer Phase

In our machine translation project the translation is done in 3 steps:

1) **Analysis**. Each word is linked to its head (one word, the root of the sentence, does not have a head). For instance, *"the boy sold a new house"* is analyzed according to Table 1.
The first column designates the word position in the sentence and the second column its head. For instance, *"house"* is the object of *"sold"*.

2) **Transfer**. The graph of words as knots and dependency relations as arches is generally a tree. The transfer phase takes all the bigrams from the root to the leaves and translates each bigram according to the transfer dictionary. First it

Table 1. Analysis result

pos	head	tag	word
1	2	DT	the
2	3	NN	boy
3	0	VBD	sell sold
4	6	DT	a
5	6	JJ	new
6	3	NN	house

tries to translate the bigram if it is declared in the dictionary; otherwise it tries to translate H or S based on the relation-tag RT of the bigram and, then, finally it tries to make a syntax rearrangement based only on RT. The next step is to translate all words that did not fit any rule for bigrams. An example of the transfer phase output is in Table 2.

Table 2. Transfer result

pos'	pos	head	tag'	word'
1	1	2	DT	o
2	2	3	NN	menino
3	3	0	VBD3s	vender
4	4	6	DTf	um
5	6	3	NNf	casa
6	5	6	JJf	novo

The translation of bigrams is done in a top-down direction because it is easier to handle the movement of words and phrases around the head-word. The statements used from the transfer dictionary are in Table 3.

Table 3. Rules

$NN << -JJ\_NN\ :\ \%+\%$
$a\ DT\ :\ um$
$boy\ NN\ :\ menino$
$house\ NN\ :\ casa.f$
$new\ JJ\ :\ novo$
$sell\ VB\ :\ vender$
$the\ DT\ :\ o$

3) **Generation**. The words must be conjugated according to their tags and the result is: *"o menino vendeu uma casa nova"*.

3.1 The Transfer Phase Internals

Each knot of our syntactic tree corresponds to one word of the sentence. If we project the design of the tree in a horizontal line then we get *proj(tree)* that is a sequence of words that corresponds to the input sentence. In the literature, the transfer phase maps one syntactic tree into another tree'. If we had adopted the same approach, then we would construct tree' and work on its projection in order to get the translated sentence. Instead of this procedure, we decided to create one structure that reflects *proj(tree)* and change it in order to create *proj(tree')* that leads to the output sentence. The structure *proj(tree')* is an array where each element *proj[pos']* corresponds to one word in the translated sentence. Each position **pos'** of this array contains (see Table 2):

pos : The reference to the English word in position pos (see Table 1). Ex: for $pos' = 5$, $proj[5].pos = 6$.

word' : The Portuguese word in its radical form. Ex: $proj[3].word' = vender$. If $proj[X].word'$ is empty (due to lack of data in the transfer dictionary) then the corresponding English word is taken as word' and put in the translated sentence.

tag' : The Portuguese tag. It is generated by rules of the transfer phase (ex: *subject-verb* and *adjective-noun* agreement) and according to the declarations in the transfer dictionary. If $proj[X].tag'$ is empty, then the English tag is used. Ex: $proj[6].tag' = JJf$.

head' : The pointer to the Portuguese head of word'. The $proj[X].head'$ is generally empty for all X in the projection. When empty, the corresponding English *head* (head of $proj[X].pos$) is used for the agreement rules in the transfer phase, otherwise $proj[X].head'$ will be used. In Table 2, we do not show the head' column because it is empty. We assume, for instance, that *head'* of *"novo"* is *"casa"* (and made the agreement in genre and noun) because $proj[6].pos = 5$ and the head of 5 in the English sentence is *"house"* that corresponds to *"casa"*.

We operate on the projection of tree' when applying the rules in the transfer dictionary. Initially, we set *proj(tree')* to *proj(tree)* by making $proj[i].pos = i$ for $1 \leq i \leq length(InputSentence)$. Then we apply the rules from the dictionary. The main rules that modify proj(tree') are:

a) Order declaration. When it is necessary to change the order of some bigram (H,S), then we:

1) Move the head H to the position of the subordinated-word S: $proj[H].pos$ becomes $proj[S].pos$.

2) Move S and all its descendents (the tree that has S as root) to the position of H.

b) suppression of S' or H'. It is just to remove one position of proj(tree').

c) insertion of word. In the rule:

$H\ H^T\ :\ H' < -P\ S$

it is necessary to locate X that is one position before (or after depending of H) of the boundary of the tree that has S as its root and make:

$proj[X].word' = P$

The other elements (pos, tag', head') can be filled if it is necessary to conjugate P.

Generally, P is a preposition that appears on the left of S and is not conjugated, therefore, nothing is set to the other elements.

4 Inside the Transfer System

Besides the transfer phase, the transfer dictionary has been applied in other phases of our machine translation project:
a) **parsing**. Bigrams have been applied to solve ambiguities during the analysis phase[3]. In our project, bigrams are used to solve compound names and preposition attachments.
For instance, the phrase *"data flow"* can be thought as a sequence of a noun and a verb (NNS VBP) and incorrectly analyzed, and translated as *"dados fluem"*. As technical texts have a high frequency of compound names, we decided to study and solve the compound names before the parsing process by just looking up bigrams in the transfer dictionary.
b) **text revision**. In our system, the information about the syntactic tree and its projection is not discarded. During the text revision, the user can modify the dictionary providing new information about words and bigrams. This new information is automatically applied only where it fits in the text. This process is made in a interactive way by the web interface:
1) The user reads the text, finds some wrong translation, and points to the word that is inside some sentence (using the mouse).
2) The system looks the word and its head from the syntactic tree and asks for a correction in the the word or its associated bigram, presenting all the possible entries of the transfer dictionary (word, bigram or partial bigram) in a form (even though they are not declared).
3) The user fills the form.
4) The system makes the correction in all the other sentences looking a cross-reference table.
This process speeds up the translation process because each revision (ex: a new translation for a word) can affect the entire text.

5 Conclusion

Operations such as exchanging word places, suppressing/inserting words are very common in the transfer phase. In spite of it, there are few works in the literature about this theme. For instance, [8] is an interesting work about the English to Japanese translation and presents a transfer grammar that is more complex than the dictionary presented here. The main contribution of this paper is to propose a division between general transfer rules that operates on the syntactic tree and transfer rules that operates only on bigrams. Simpler rules for bigrams were possible because, even though two words in one language can be translated as a different number of words in another language, our notation enables a bigram to be always mapped as another bigram. This approach can not cover all transfer rules. For instance, the trigram *"as well as"* requires other rules. This

formalism is being applied to our machine translation project with success. The dictionary enables a simple transfer system, for instance, it is not necessary any complex grammar for sentence generation. We do not map a syntactic tree into another one. In this approach, it is necessary to match a small tree (the rule) into the syntactic tree. Instead, we change the projection of the tree observing only bigrams, which simplifies the sentence generation.
The simplicity of the formalism suggests that it could be applied (or easily adapted) to projects that has an analysis similar to that provided by dependency grammars. The parser based on the Link Grammar [7] builds a planar graph (the words are knots, and the arches are the links between 2 related words) for each sentence that could be analyzed. In this graph, each arch has a label (our relation-tag) that provides syntactic information. The dictionary presented in this paper can be easily adapted to the Link Grammar work by the tasks:
1) transform a planar graph in a tree or forest of trees (sometimes it is necessary to cut some linkages).
2) adapt the nomenclature of relation-tags.
3) apply the dictionary as it was shown in this paper.
The project is in evolution and we intend to make the transfer dictionary work with the WordNet [5]. It is expected that it is possible to have just one translation for each meaning in WordNet.
The transfer dictionary notation works fine due to the great tendency in human languages to group related words together as it was observed in many theories ([9], [10], [4]). Some cases can not be handled by bigrams (subject-modal-verb agreement that requires a trigram, the treatment of the particle "TO", idiomatic expressions, etc.) and are treated by other modules in the transfer phase.

acknowledgment : partially supported by FAPESP.

References

1. KINOSHITA, J.: Aspectos de Implementação de Uma Ferramenta de Auxílio à Tradução Inglês-Português. Tese de Doutorado, Departmento de Computação e Sistemas Digitais da Escola Politécnica da Univesidade de São Paulo, Brazil (1997)
2. DOUG, A.: Transfer. working paper, Department of Language and Linguistics, University of Essex, UK, (1997) 1–8
3. HINDLE,D. MATS,R.: Structural Ambiguity and Lexical Relations. Computational Linguistics, v. 19, n. 1, (1993) 103-120.
4. MELCUK, I. A. : Dependency syntax: theory and practice. New York, State University of New York Press, 1988. apud SLEATOR (1991)
5. MILLER, G.A. ET AL.: Introduction to WordNet: an on-line lexical database. Princeton University, http://www.cogsci.princeton.edu/ wn/, (1993)
6. PENN TOOLS. http://www.cis.upenn.edu/ adwait/penntools.html, (1998)
7. SLEATOR, D. D. K.; TEMPERLEY, D. : Parsing English with a Link Grammar. Pittsburgh, Canegie Mellon University, sleator@cs.cmu.edu (1991)

8. TSUTSUMI,T.: Wide-range restructuring of intermediate representations in machine translation. Computational Linguistics, v. 16, n. 2, (1990) 71–78.
9. RADFORD, A. : Transformational grammar: A first course. Cambridge, Cambridge University Press, (1989).
10. RAPOSO, E.P. : Teoria da gramática: A faculdade da linguagem. Lisboa, Editorial Caminho S.A., (1992).