Translation of verbal idioms

Martine Smets, Joseph Pentheroudakis and Arul Menezes
Microsoft Research
One Microsoft Way
Redmond, WA 98052, USA
martines@microsoft.com
josephp@microsoft.com
arulm@microsoft.com

Abstract

Verbal idioms constitute a challenge for machine translation systems: their meaning is not compositional, preventing a word-for-word translation, and they can be discontinuous, preventing a match during tokenization. This paper presents the treatment of verbal idioms in our machine translation system, which addresses both challenges by deferring idiom matching until after the parse, and by allowing all parts of an idiom to be treated as a lexical unit, making alignment and transfer easier. Identification of idiom use is not a problem for us, because we learn idiom translation in context.

1 Introduction

We distinguish verbal idioms (idioms headed by a verb) from fixed multiword entries: the latter, in our definition, cannot be discontinuous, and are stored like ordinary words in our monolingual dictionary. Examples include nouns such as *mise à jour* ('update') and *prise en charge* ('support' in our technical corpus), conjunctions such as *de sorte que* ('so that'), and prepositions (*à la fin de* 'at the end of'). These expressions do not require a specific treatment, and are handled in the same way as single words of the same category.[1]

For the purpose of this paper, we define verbal idioms as verb phrases whose meaning is idiomatic and cannot be derived compositionally from the literal meaning of the idiom parts. Verbal idioms thus pose problems for natural language systems, and especially machine translation systems, where the entire phrase may have a non-compositional gloss. For example, in the system presented in this paper, the French idiom *prendre en charge* has been variously translated word for word (and therefore incorrectly) as 'take in load' or 'seize in load', when the correct translation in our technical context is 'support'.

Other examples include *faire partie de* ('belong'), translated by 'make part' in some instances, *arriver à expiration* ('expire'),

[1] It was pointed out to us that it is actually possible to come up with examples where multiword entries are not contiguous, as in *à la fin, dit-il, de cette histoire.* However, we have not come across such cases in our technical corpus; if we did, the analyzer would simply analyze the phrase compositionally instead of treating it as a single entry.

translated by 'get expiration', *avoir besoin de* ('need') translated by 'have need', etc. These verbal idioms are very common, and are typically translated very poorly. The problem in all these cases is that these verbal idioms are not analyzed as such, and are translated literally, word for word.

In this paper we present the approach we have implemented to achieve a better translation of verbal idioms.

2 Verbal idioms

We discuss in this paper idioms referred to as 'idiomatically combining expression' in Nunberg et al (1994). Their idiomatic meaning is compositional, in that it is possible to determine which part of the idiom carries which part of the idiomatic meaning. Nunberg et al. oppose this type of idioms to what they call 'idiomatic phrases', idioms whose idiomatic interpretation cannot be distributed over their parts, such as the much quoted *kick the bucket.* We will not be discussing this last type of idioms in this paper, although the treatment we are going to propose could be extended to them.

Verbal idioms can participate in a variety of constructions, which can result in discontinuities, and vary according to the idioms (Nunberg 1994, Schenk 1995, Wehrli 1998 among others). That, in turn, makes it difficult to match all parts of the idiom in a sentence. For example, in the examples below, the object is not adjacent to the other idiom parts because of relativization (as in (1)) and the passive construction (in (2)):

(1) Les affaires qu'il a demandé à prendre lui-même en charge sont délicates.

(2) Ces affaires seront prises en charge sans délai.

The entire expression has to be recognized as a unit, however, if translations such as the ones mentioned in the introduction are to be avoided. For the expression to have an idiomatic reading, *prendre* must be followed by, although not necessarily be adjacent to, the prepositional phrase *en charge*; it must also have an object complement (which, in the passive construction, will be realized as the grammatical subject).

Previous approaches to idiom analysis propose to identify idioms during parsing (for example, Stock 1989 and Matsumoto et al. 1991), or on the structure produced by parsing (Wehrli 1998). Some approaches propose local grammar rules written specifically to handle idioms (Breidt et al. 1996).

Our approach is closest to Wehrli's solution, in that idioms are identified after parsing (in our case, on the resulting syntactic tree). As we pointed out earlier, since idioms can be discontinuous, the entire sentence has to be parsed before an idiom can be identified with certainty. In our current implementation, idioms such as these are entered manually in the monolingual dictionaries. The entries are keyed on the verbal head, and they list the arguments and modifiers that make up the idiom, with morphosyntactic constraints expressed as features on each idiom part. For example, *avoir besoin de (quelque chose)* is an idiom meaning 'to need (something)' as long as *besoin* is in the singular and is not preceded by a determiner; that information is hand-coded in the dictionary. Our MT component, on the other hand, does not use hand-coded general-purpose bilingual dictionaries;[2] rather, only domain-specific bilingual dictionaries are used, automatically learned on the corpus used to train the transfer component. These bilingual dictionaries include single words or fixed multiword

[2] But it uses a small hand-crafted dictionary of function words: prepositions, pronouns and conjunctions.

expressions[3]; they do not currently learn verbal idioms with internal structure, and as a result no translation entries exist for these items.

The remainder of this paper presents our MT system and the solution we have adopted for matching and translating verbal idioms.

3 Overview of our MT System

The MT system discussed here uses a source language broad coverage analyzer, a large general-purpose source language dictionary, a learned bilingual dictionary, a small bilingual dictionary of function words (around 500 entries),

an application-independent natural language generation component which can access a full monolingual dictionary for the target language, and a transfer component. The transfer component, described in detail in Menezes & Richardson 2001, consists of high-quality transfer patterns automatically acquired from sentence-aligned bilingual corpora.

We will describe the analysis and the transfer module in some detail, since that is where verbal idioms are matched and then translated, respectively.

3.1 The analysis module

Analysis produces three representations for the input sentence: sketch, portrait and logical form[4] (Jensen 1993, Heidorn 2000).

Sketch is the initial tree representation for the sentence, along with its associated attribute-value record structure. The grammar is an augmented phrase structure grammar, whose binary rules are constrained by conditions on the combined constituents and actions on the resulting constituent (Jensen 1993). Syntactic rules produce at least two structures, a derivation tree (with binary branching) and a computed tree (a structure flattened on the basis of the syntactic information contained in the derivation tree).

An example of a computed tree for a sketch analysis is given in Figure 1. It is the tree representation for the sentence in (3).

(3)
Ce format est pris en charge par Windows 2000
This format is taken in charge by Windows 2000
This format is supported by Windows 2000

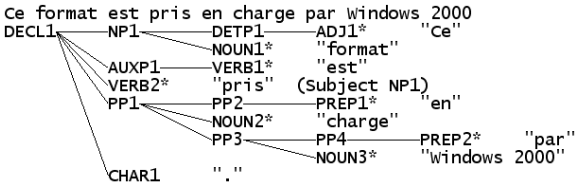

Figure 1 : Sketch

Attachment sites for post-modifiers are not determined in sketch. In most cases, the information available when the syntactic tree is being built is not sufficient to determine where prepositional phrases or relative clauses should attach. Post-modifiers are thus systematically attached to the closest possible attachment site. Therefore, the sketch needs to be further processed by the reattachment module, which produces the *portrait* analysis for the sentence (Jensen 1993, Heidorn 2000).

The reattachment module is a set of rules which consider several possible attachment sites for certain types of phrases: for example, post-modifying prepositional phrases, adjective phrases (for French), or relative clauses. These rules are heuristics only, and assign scores to competing attachment sites. A phrase is moved to a

---

[3] This learned bilingual dictionary includes multiword expressions encoded by hand in monolingual dictionaries and menu names, titles, etc., put together prior to parsing by the morphological component.

[4] The presentation of the analysis module is somewhat simplified, but sufficient for our current discussion of idioms. More details can be found in the references.

new site as specified by the rule with the highest score. An example of reattachment is given in Figure 2. The PP constituent expressing the agent of the passive construction has been reattached to the head (the verb).

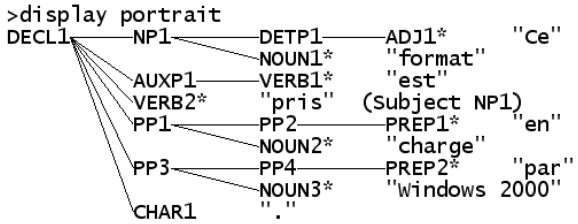

Figure 2 : Portrait

The third stage in analysis is the computation of the logical form (LF). The logical form is a labeled directed unordered graph which contains the deep syntactic structure, such as predicate-argument structure, and some semantic information for the input sentence, for example the relations expressed by prepositions. At this level, the difference between active and passive constructions is normalized (they have the same logical form); also, control relations and long-distance dependencies are resolved (subjects of infinitives, arguments associated with gaps, etc.). An example is given in Figure 3, the logical form of the sentence *le fichier de commandes que vous voulez traiter* ('the batch file that you want to process'). The controller of *traiter* has been determined, as has the relation between *fichier de commandes* and the gap in the relative clause.

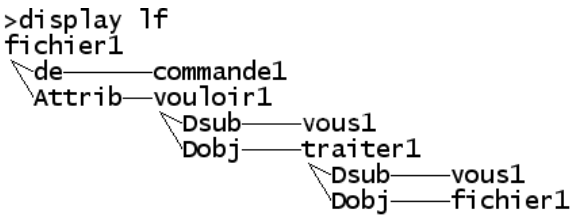

Figure 3 : Logical form

The rules that compute the logical form map constituents to semantic roles or grammatical functions; additionally, the distinction between pre- and postmodifiers of a given head may be lost. Note, however, that logical form rules do not select attachment sites for modifiers, for example; that is the province of the reattachment component. In other words, attachment ambiguities need to be resolved in the input to the logical form component.

3.2 The transfer component

The transfer component consists of a training phase and a translation phase. The training phase learns transfer mappings from a sentence-aligned bilingual corpus. During training a LF alignment algorithm is used to align source language and target language logical forms at the sub-sentence level. The LF alignment algorithm first establishes tentative lexical correspondences between nodes in the source and target LFs using translation pairs from a learned bilingual lexicon and a small hand-crafted bilingual lexicon of function words. After establishing possible correspondences, the algorithm uses a small set of alignment grammar rules to align LF nodes according to both lexical and structural considerations. The aligned LF pairs are then partitioned into aligned LF fragments that comprise the transfer mappings. The final step is to filter the mappings based on the frequency of their source and target sides. Menezes & Richardson (2001) provides further details and an evaluation of the LF alignment algorithm.

During the translation phase, we search the transfer mappings acquired during alignment, for mappings that match portions of the input LF. We prefer larger (more specific) mappings to smaller (more general) mappings. Among mappings of equal size, we prefer higher-frequency mappings. We allow overlapping mappings that do not conflict. The lemmas in any portion of the source LF not covered by a

transfer mapping are translated using the learned bilingual dictionary. We then stitch these mappings and dictionary translations together into a target language LF. From this, a rule-based generation component produces the target language sentence.

Information added to logical forms by the idiom matching algorithm is used by the transfer component to pair idioms of the source language to appropriate translations in the target language. How idioms are matched and mapped with appropriate translations is discussed in the next sections.

4 Idiom Identification

As we mentioned above, verbal idioms are assigned their own entries in the monolingual lexicon. These entries specify the arguments, modifiers, and other morphosyntactic attributes that need to be present in the input for a given verbal idiom to be identified as such. The entry for the idiom prendre qqch en charge is shown below:

```
{Word       "prendre_qqch_en_charge"
 Verb
       {Lemma  "prendre_qqch_en_charge"
        Key    prendre
        Args
               {Cat        NP
                Bits       Acc}

               {Lemma      "charge"
                Bits       Sing
                Cat        PP
                Prep       "en" } } }
```

The attribute *Args* lists the arguments and modifiers of the idiom. The order of the arguments in the entry is not significant in our current implementation. The notation *Bits Acc* in the NP record indicates that the case of a matching NP should be accusative, which reflects our representation of direct object NPs.

The idiom matching algorithm is invoked after the initial analysis and before reattachment. The algorithm itself does not modify the parse; rather, it simply searches the tree for constituents that match the arguments and modifiers of any idiom entries associated with each verb in the tree. Remember that the sketch for a sentence like *il a pris ses affaires en charge* will show the prepositional phrase *en charge* as modifying the noun *affaires*:

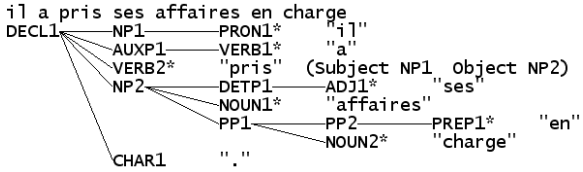

Figure 4 : Sketch

Logical form rules, however, expect attachment ambiguities to have been resolved. In this case, then, the prepositional phrase *en charge* needs to be promoted from its position as a modifier of *affaires* to modify *prendre*. Idiom matching to the rescue: the algorithm walks the tree searching for constituents that match the lexical specifications for the idioms associated with *prendre*. If any such constituents are found, each is co-indexed as being a part of that idiom, regardless of whether all parts of the idiom are found. In the event that *all* parts of an idiom are found during the search, the idiom record for the verbal head is marked as being fully saturated, or *Completed*, in our notation.

Note that if the sentence were simply *il a pris ses affaires*, the idiom matching algorithm would identify *ses affaires* as matching the direct object NP slot of the idiom *prendre (quelque chose) en charge*; however, since the prepositional phrase *en charge* is not present in the sentence, the idiom will not be marked *Completed*. This allows matching to proceed incrementally, deferring the decision on whether the entire idiom is present until the end.

In searching for idiom parts, the algorithm needs to follow only syntactically valid paths. For example, in the sentence in (4), the algorithm should not match the PP *en charge* as part of the idiom, since that PP is not within the scope of the head verb *prendre*.

(4) Puisqu'il était en charge de cette affaire, il a pris sur lui de convoquer la presse
'Because he was in charge of this matter, he took it upon himself to invite the press'

To ensure that only syntactically valid paths are searched, the algorithm invokes language-specific rules specifying those paths.

Finally, it may be possible that the parts matched by the algorithm are already in the proper relation with respect to their head, and that therefore no reattachment is necessary. For example, there is no attachment ambiguity in the parse for the sentence *elle a peur* 'she is scared', using the idiom *avoir peur* 'be scared, be afraid'. Consequently, the reattachment rules will not modify the tree:

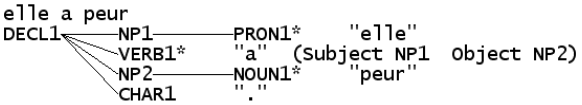

Figure 5 : Sketch and portrait

This illustrates the point that the primary purpose of the algorithm is simply to search for and co-index the parts of an idiom, and to indicate whether or not an idiom was fully instantiated. Reattachment may use that information to revise the tree. The primary purpose of the co-indexing annotations, in other words, is to ensure that other system modules can treat the idiom constituents as a single, coherent whole. Those annotations are inherited by the logical form records, allowing those rules, as well as the alignment and transfer modules, to also treat those records as a coherent whole.

The annotations also indicate whether or not a given record in the tree matched an idiom part that is not lexically constrained. For example, in the by now familiar idiom *prendre (quelque chose) en charge*, any NP can fill the direct object slot. This is reflected in the translation of that sentence; in our technical domain, *prendre X en charge* translates as *to support X*. In other words, while the co-indexing method allows us to treat the idiom parts as a whole, it is still possible to distinguish those parts that form a frozen, fixed part of the idiom from those which can be instantiated by an open class of words.

5 Idiom translation

The goal of idiom translation is to ensure that we learn, during training, a transfer mapping for the idiom as a whole, rather than literal word-for-word mappings for portions of the idiom. Once we have learned such a mapping, during translation, it will automatically be preferred over word-for-word alternatives, since we prefer larger mappings over smaller ones. Hence idiom-specific handling is required only in the training or alignment phase.

The idiom handling in alignment uses information gathered by the idiom-matching algorithm about which nodes and relations participate in an idiom. This information is numerically encoded in the LF so that every LF node that participates in an idiom has the following information.

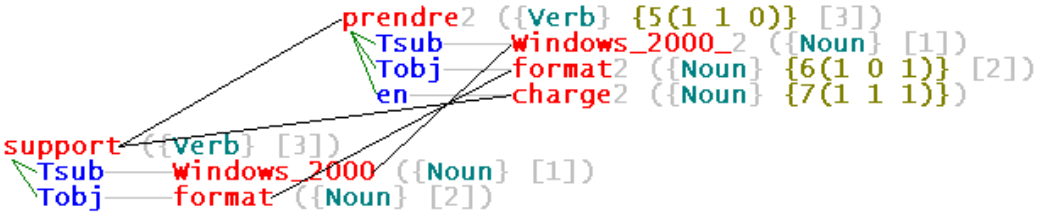

Figure 8 : Aligned logical forms

1. Which idiom the node is part of (a sentence may contain multiple idioms so each idiom is given an identifying idiom number)
2. Whether the node is the head of the idiom
3. Whether the lemma of the node participates in the idiom
4. Whether the relationship between two nodes participates in the idiom

Figure 6 shows the logical form for the example in Figure 1. In this example, the nodes *prendre* and *charge* participate in the idiom, as do the relation expressed by *en* and the *Tobj* relation between *prendre* and *format*. However the lemma *format* is not part of the idiom. Finally the node *prendre* is marked as being the head of the idiom.

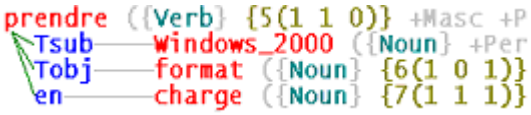

Figure 6: Logical Form with idiom encoding

During alignment, the alignment grammar uses this information to ensure that idioms are always aligned as a whole. We illustrate this with an example using the sentence pair in (3). Figure 7 shows the logical forms for these sentences.

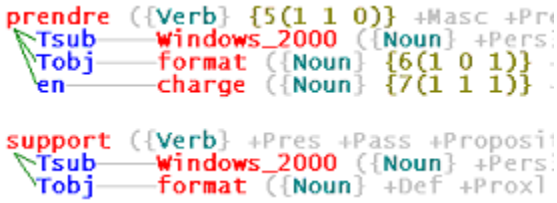

Figure 7: Logical Form pair to be aligned

The alignment grammar uses lexical and structural correspondences, and proceeds in a "best-first" manner, aligning the strongest correspondences first, and using these to align weaker correspondences (see Menezes & Richardson, 2001 for details). In this example, *Windows_2000* and *format* are aligned first based on lexical correspondence. The verbs *prendre* and *support* are aligned next, based purely on structural grounds, namely that their respective subjects and objects have already been aligned to each other. The system has no pre-existing knowledge of any lexical or semantic correspondence between *prendre* and *support*. After aligning *prendre* and *support,* the alignment algorithm notes that *prendre* is part of an idiom. It therefore includes the rest of the idiom in the alignment, resulting in *charge* also being aligned to *support*. The resulting alignment is shown in Figure 8. Note that *prendre* and *charge* are both aligned to *suppor*t.

From these aligned LF pairs we acquire transfer mappings, by dividing the aligned LF pairs into multiple overlapping LF fragments. In this phase we use the idiom information encoded in the LF as described above to ensure that all the lemmas and relationships that are part of the idiom are not partitioned into separate mappings but stay together as a whole.

For example, the *Tobj* relation must be an attribute of *prendre* because it is part of the idiom, even though the actual object *format* is not. Figure 9 shows some of the mappings derived from this aligned LF pair.

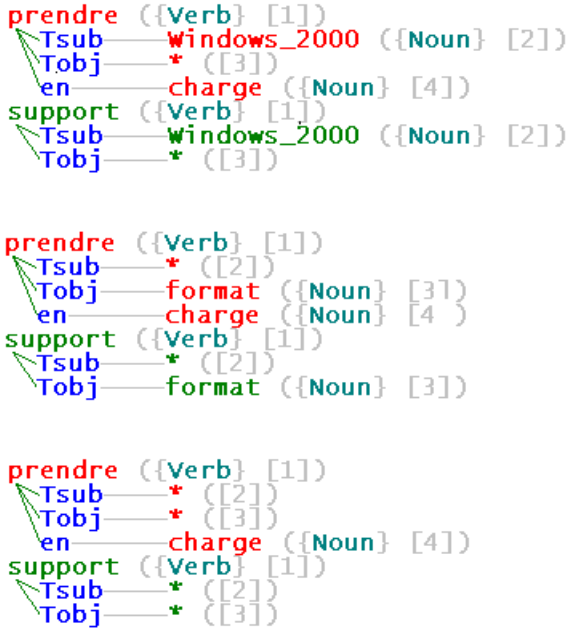

Figure 9: Transfer mappings

6 Results and future work

We have tested our approach on a corpus of around 1000 sentences from Microsoft technical manuals. However, so far we only have a handful of idioms encoded in the French monolingual dictionary. Among the verbal idioms stored in the French dictionary are *prendre en charge, arriver à expiration, mettre à niveau, mettre fin à, mettre à jour.* The translation of these verbal idioms improved after we implemented our approach. For example, *prendre en charge* is now translated only by 'to support'. Similarly, the translation of *arriver à expiration*, which used to be 'get expiration' is now 'expire', and *mettre à niveau* is now translated by 'to upgrade' instead of 'to put at level'. As for *mettre à jour*, it is translated in most cases as 'to update', but in one case as 'put updated' (*à jour* is translated by 'updated' in some instances of our corpus).

Since translation of idioms is learned automatically in the domain which is being translated, the problem of whether a collocation is used idiomatically or literally does not arise. Idiomatic use depends on the context (morphosyntactic, syntactic, semantic), and context is what the transfer component relies on to learn mappings.

These are thus encouraging results, but more needs to be done to evaluate our approach. We are planning to first enrich our monolingual dictionaries, using automatic learning. We will be implementing techniques for detecting and learning verbal idioms automatically from corpora, since different domains may use different idioms.

We are also planning evaluation of our approach on another corpus: the Hansard parliamentary texts. That corpus is much richer in verbal idioms, and should be a more interesting test case than the technical corpus.

7 References

Breidt, E., Segond F and Valetto G. (1996) "Local grammars for the description of multi-word lexemes and their automatic recognition in texts", *Proceedings of COMPLEX96*, Budapest.

Heidorn G. E. (2000) "Intelligence writing assistance", in Dale, R. , Moisl, H. and H. Somers (eds) *Handbook of Natural Language Processing.*

Jensen, K. (1993) "PEG: the PLNLP English grammar", in Jensen, K., Heidorn, G. and S. Richardson (eds) *Natural Language Processing: the PLNLP Approach*.

Matsumoto Y, Yamagami K and Nagao M. (1991) "Bi-directional parsing for idiom handling", in Martin, J., Fass, D. and E. Hinkelman (eds) *Proceedings of the IJCAI Workshop on Computational Approaches to Non-Literal Language: Metaphor, Metonymy, Idioms, Speech Acts and Implicature*.

Menezes, A. and S. Richardson (2001) "A best-first alignment algorithm for automatic extraction of transfer mappings from bilingual corpora", *Proceedings of the Workshop on Data-Driven Machine Translation, ACL 2001*, Toulouse, France.

Nunberg, G., Sag, I. and Wasow, T. (1994) "Idioms", *Language,* 70:3.

Schenk, A. (1995) “The syntactic behaviour of idioms”, in Everaert, M., van der Linden, E., Schenk, A. and R. Schreuder (eds) *Idioms: Structural and Psychological Perspectives*.
Stock, O. (1989) “Parsing with flexibility, dynamic strategies, and idioms in mind”, *Computational Linguistics*, 15.1.
Wehrli, E. (1998). “Translating idioms”, *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics*, Montreal.