

Handling Multi-word Expressions without Explicit Linguistic Rules in an MT System

Akshar Bharati, Rajeev Sangal, Dipti Mishra, Sriram V., Papi Reddy T.

International Institute of Information Technology – Hyderabad

Email: {sangal,dipti}@iiit.net, {sriram,papi_reddy}@students.iiit.net

Abstract. Translation of Multi-word expressions (MWEs) is one of the most challenging tasks of a Machine translation (MT) system. In this paper, we present an innovative technique for dealing with MWEs in the context of MT. The technique permits bilinguals to give translations of MWEs in the form of patterns, without requiring them to be trained linguistically. The interpretation of the patterns is done by a dynamic machine learning algorithm, which allows the main rule-based MT system to operate based on linguistic rules. Thus, the bilingual patterns (without any explicit linguistic input) are used in conjunction with the main linguistic system. This is made possible by the learning pathway templates. These templates need to be specially prepared by trained linguists only once. After that they help to process potentially a large number of patterns.

The implemented system is being used with a large-scale rule-based MT system to improve its performance. This framework can also be extended to help example-based or statistical MT systems to deal with MWEs.

1 Introduction

This paper addresses the problem of developing a technique for handling MWEs in the context of a rule-based MT system.

MWEs are expressions with a special meaning, which cannot be derived from its component words. MWEs include, among others, idioms ('kick the bucket' instead of 'die'), phrasal verbs ('carry on' instead of 'continue'), and compounds ('judicial enquiry'). A typical natural language system assumes each word to be a lexical unit, but this assumption does not hold in case of MWEs. They have idiosyncratic interpretations that cross word boundaries. Thus, identification and generation of MWEs has been a major concern for scholars working in this area and these are, therefore, considered a 'pain in the neck' (Sag et al., 2002).

Even though, several of these MWEs are not compositional semantically, they behave like any other phrase syntactically, i.e., they take inflections, modifiers etc, and undergo syntactic operations such as passivization etc. Therefore, when it comes to translating such MWEs it becomes all the more complex since, after identification, they need to be processed linguistically. Their corresponding target language equivalents also need to be generated. Hence, a large dictionary is required to better the performance in translation but it often becomes a bottleneck as building dictionaries is not an easy task. It requires immense amount of time and effort. It is not always possible to either automatically generate this data or have language experts to develop this.

On the other hand, ordinary bilinguals are a rich resource of such data. They can provide parallel expressions in the two concerned languages. If one can tap on this resource, sufficiently large amount of data can be prepared in a shorter time. Though, the data thus created would lack linguistic knowledge necessary for analysis. It cannot, therefore, be used directly, for processing in a rule-based MT system. However, if a mechanism can be developed to interpret the data linguistically and then use it in conjunction with the rest of the linguistic processing, this can be an effective approach for handling MWEs in an MT system.

The present paper is an attempt towards evolving a mechanism whereby the MWEs are incorporated in the main linguistic processing system through learning pathway templates. The idiom dictionary itself is developed by bilinguals who need not have any linguistic training. A simple notation is developed using which non-linguists can specify patterns for MWEs. These patterns are connected to the main linguistic system using Learning Pathway Templates. The concept of templates is interesting and can be generalized in other kinds of learning templates.

This system is implemented and is being used with a large scale English to Hindi MT system.

2 General Problem

MWEs are special constructions that require appropriate representation and analysis. Several of them show lot of variation (Segond & Tapanainen, 1996). However, there is much in MWEs that is mechanizable and computable.

An example of a MWE of this type is,
 "simmer with anger" (1)

The above MWE can undergo the following types of variations.

2.1 Morphological Variation

'Simmer' can be inflected for tense, aspect and modality, e.g., simmers, simmering, simmered. Similarly, nouns in the MWEs can occur in varying forms depending on the gender, number etc.

2.2 Insertions

New words (qualifiers) can be inserted within the MWEs, as in:

"(sadly) simmering with (quiet) anger" (2)

OR

"has simmered (for quite a while) with (quiet) anger." (3)

2.3 Replacement

In an MWE, one word can replace another without affecting the overall characteristics of that MWE. For example,

"simmer with [fear]" or "simmer with [pain]" have similar characteristics as (1). The words in [] are words that took the place of the actual word.

The MWEs obey and fit inside the linguistic framework of the language (Wehrli, 1998) i.e., though these MWEs have a non-compositional semantics, structurally, they behave like any other phrase in a sentence with various parts of the expression taking their inflections, allowing modifications etc. Clearly, processing such expressions is not easy. While handling MWEs, we have to combine the specialized processing of MWE, with the usual processing of a sentence. The two processes would ensure that the non-compositional part is taken as a unit, and processed further using the usual mechanism of producing meaning out of compositional parts.

Identifying such MWEs and providing their meaning can be done by people. As mentioned above, we have found that people familiar with the application can provide the meaning without necessarily having elaborate linguistic training. For applications such as machine translation, the number of MWEs to be handled for good quality translation is extremely large. Therefore, it is crucial to be able to use the contributions of large number of bilinguals without requiring that they learn linguistics first. A problem, with the data thus created, as mentioned earlier, would be how to integrate it into the larger linguistic system since it will lack the necessary linguistic information. As illustrated by some examples earlier (simmer with anger), unless the two are combined, the coverage provided by the MWE data would be miniscule.

The proposed solution is to couple the data with the linguistic system. This coupling can be done using the learning pathway templates. The above also fits in with the machine learning of rules, etc, in case we want the machine to "abstract" and generalize from the data thus provided. In fact, machine learning becomes possible with much smaller amount of data, because the data already fits in with the linguistic framework and the generalizations can work along well-defined pathways.

3 An overview of related work

Before proceeding with this approach a brief mention of related work is presented.

Segond and Tapanainen's work on 'Using a Finite-state based Formalism to Identify and Generate Multi-word Expressions' (Segond & Tapanainen, 1996) demonstrates how a multi-word expression can be encoded, and how their compiler would use them to identify the MWEs. In contrast, our formalism is simple, yet expressive. The person providing the MWEs can provide the data in a most natural way. The system learns the linguistic characteristics of the MWE on its own, using the learning pathway templates. The system then uses the learned patterns to identify MWEs in a sentence.

Wehrli's work on translating idioms (Wehrli, 1998) talks about how MWEs can be used by a linguistic system. It also talks about the transfer and generation of idioms in its framework. Our approach of generation is similar to Wehrli's approach. Our framework introduces the concept of compiled pattern, which is used to do the generation robustly.

The MWEs, which can be collected using our framework, can also be induced into the example base of a lexical EBMT system (Brown, 1999). Every MWE can be represented as a separate equivalence class (token). The translation of this token can be remembered as the 'substitution string' (suggested in this framework).

4 Framework

4.1 Patterns for MWEs

The setting of our application is a rule-based MT system, which is already available. Its coverage might be low, or the output quality poor. A typical reason could be that it does not handle MWEs. The outputs produced by the system are looked at by bilinguals (call them language editors) who are not trained in linguistics. They correct the translation and in case they find that the error is due to an MWE, they also provide the MWE and its translation. They are also encouraged to provide simple patterns to cover a class of MWEs. This, however, does not require them to provide linguistic analysis.

Each input provided by the language editors consists of two items:

1. Example sentence (with a MWE) in source language and its translation in the target language.
2. MWE pattern in the source language and its translation (the MWE in the pattern must occur in the example sentence).

It is a requirement that each pattern must be lexicalized, which means that there must be at least one lexical item associated with the pattern.

An example pattern (P1) for English to Hindi MT system looks as follows:

Pattern P1:

1. Example sentence and its translation: (4)
 Godhra is simmering with anger
 Godhraa krodha se bhabhak rahaa hei
 (anger) (INSTR) (heated_state)(ing)
2. MWE pattern and its translation (5)
 Simmering with anger*1
 krodha*1 se bhabhak rahaa hei
 (anger) (INSTR) (heated_state)(ing)
 *1 = anger, frustration, pain

4.2 Notations

Variables in the pattern are marked by '*'. *1 in example (5) says that anger is a variable and can be replaced by any of the words anger, frustration or pain. If a list of words is not given, then it can be replaced by any other word of same category, which is noun in example (5). *1 also associates anger to krodha in the target language.

In this pattern, all the inflections of a word are allowed. Thus, the example pattern (4) would allow simmering, simmers, simmered etc. In case of verbs, auxiliary verbs can also be added; for example 'has been simmering' (As we will see shortly, it is the linguistic analysis that makes it possible.) If the user wants to disallow other forms of a word, he can put '!' to indicate this .

For example in,

Regret to tell! ⇒ bataate hue dukha haiki
 (while eating) (sad) (is that)

The '!' symbol says that the form of tell is fixed and cannot occur as: tells, telling etc. No auxiliary verbs are allowed either. Tell itself can be generalized to any word with the same lexical category, in other words, any verb, such as say, eat, can occur instead of tell but the form must remain the same.

Operators	Root	Category	Other features
*	X	T	X
!	T	T	T
Not(*,!)	T	T	X

4.3 Learning pathway templates

Learning pathway templates connect the patterns to their linguistic analysis. These meta-patterns are specified by experienced linguists working on the MT system. Once specified, they are used by the MT system to linguistically interpret the patterns given by the language editors, and to use the patterns appropriately while translating.

The templates are small in number and each covers a set of patterns, which have the same linguistic analysis. They specify the head of the MWE, associations from the source language pattern to the target language pattern and also give the agreement between different components of a pattern.

For example, here is a learning pathway template:

Template T1: (6)
 VG& PP ⇒ PP VG&
 {tgt_vibh='INSTR'}

T1 states that some of the patterns given by the language editors consist of two components: VG (for verb group) followed by PP where VG is the head of the pattern (marked by '&'). It further says that in the translation, the order of the two components is reversed (shown after '⇒'). It also specifies the value of a feature called tgt_vibh (namely, the case ending in target language output).

4.4 Compiling the Patterns using Templates

We now illustrate the process by which an MWE pattern may be compiled for future use, utilizing the templates. It can be done in two ways:

Using the user patterns: Parts-of-Speech taggers and chunkers are used to assign POS-tags to words and to group them into chunks in the example sentences (given along with a pattern) in source and target languages. The tag and chunk information from the sentences are induced in the pattern. For a word not marked by '*' or '!', in a given pattern, only roots and lexical category are kept, other features such as gender, number, person and tense, aspect, modality are dropped.

For words marked by '*', root is also dropped, only lexical category is kept. For words marked by '!', other features are retained irrespective of root. For example, the pattern P1 looks as follows after the induction process:

((simmering)) [[with N*1]] ⇒

VG PP
 [[N*1 se]] ((bhabhaka))
 PP VG
 Heated state

Next, the patterns with chunks are matched with templates and the specified heads and features are transferred from the matched template to the pattern.

After this processing using template T1, the processed pattern P1 looks as follows.

(P1-T1):
 ((simmering))& [[with anger*1]] =>
 [[krodha*1 se]] ((bhabhaka))&
 {tgt_vibh='INSTR'}
 PP VG

*1 = anger, frustration, pain, & marks the head.

Note that feature tgt_vibh gets transferred from template T1 to the compiled pattern. The target language pattern is also compiled into the substitution string. For example, we get after compilation:

*1_se_bhabhaka

where *1 indicates that an element corresponding to the variable is to be substituted followed by case marker 'se' and verbal root bhabhaka. Note that the order of elements is as given in the RHS of the template, which means that the reordering would get done as, specified by the pattern while generating the above compiled string.

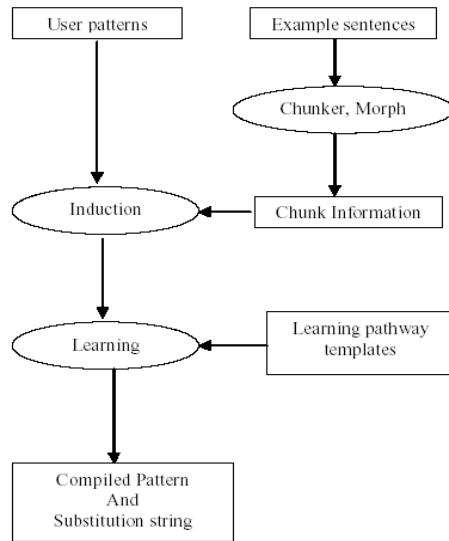


Fig.1. Compiling the pattern

Using the analyzed patterns This approach is similar to the 1st approach. The only difference is that here we receive a dictionary of MWEs that has already been analyzed by linguists and we proceed from there. This data is readjusted automatically to make it compatible with the chunks that the chunker forms. It can then be processed in the same way as the 1st approach i.e., it is matched with the learning pathway templates to get the compiled pattern and substitution string.

4.5 Processing MWE in an Actual Sentence

The MT system first does chunking and the morph analysis of the given input sentence for machine translation. This identifies root, part-of-speech tag and other features for each of the words, besides grouping them into chunks. Now, the root of words in an input sentence is matched with the compiled patterns. The process is efficient because the lexical items in the pattern are used in matching with the given input sentence.

For example, after chunking the given input sentence,

“Godhra was simmering with quiet pain”

We get the following matches of roots or words (marked by '#');

[[Godhra#]] ((was simmering#)) [[with quiet pain#]]
 NP VG PP

The above step generalizes from language elements in patterns to linguistic structures and is a crucial element in processing. This generalization is made possible with the help of taggers and chunkers which are used while processing an input sentence and the learning pathway templates which were used in the compilation of the patterns. Indeed the name pathway was chosen because it specifies a path leading from language data to linguistic theory. The pathway templates were used in compiling the templates. The same pathway can also be used if we try to generalize out of large collections of patterns when operating in a purely learning mode (though not discussed here). After the step discussed above, the processing procedures in the usual way by the linguistic processing MT system. Therefore, adverbs, adjective, etc intervening between the matched chunks are handled without any problems. In other words, the benefits of linguistic processing are available even though the MWE is being processed in a special way. Finally, the substitution of the target language expressions is done in a special way. All the chunks except the matched ones are substituted by the target language expressions in the usual way. For the matched chunks, no substitution is done for the non-head chunks. The matched chunks marked as the head (as specified by the pathway template) is substituted by the compiled target string. The Fig. 2. illustrates the substitution, where *1 is obtained by translating in the usual way by the linguistic system, i.e.,

[[with quiet pain]] \Rightarrow shaanta darda

which is then substituted at the appropriate in the compiled target expression.

Finally, the appropriate inflections are generated, yielding actual word forms. For the matched chunks, the pathway template may overwrite and specify its own values. While generating the inflections, it uses the case endings (vibhakti) and other features in the usual way without any special way for MWEs or for non-MWEs. For example, the following gets generated;

Godhraa shaanta darda se bhabhaka rahaa hei
 (Godhra quiet pain with heated state-ing)

Where 'Godhraa', the subject, agrees in gender, number, person with the verbal form of 'bhabhaka', and tense-aspect-modality of the verb is obtained from the English sentence ('ing'), as it would be done for any other verb. Note that the adjective quiet appears at the right place even though MWEs did not mention anything about adjectives. Similarly, the auxiliary verbs are produced correctly and at the right place. All this is the result of combining the linguistic knowledge with the MWE patterns. Fig. 3 summarises this whole process.

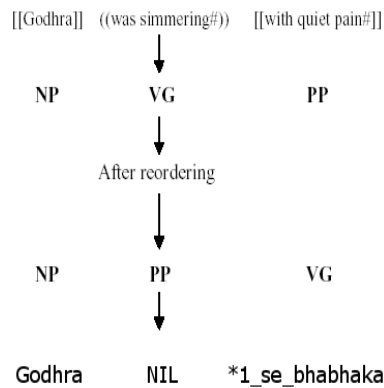


Fig. 2. Processing an Actual Sentence

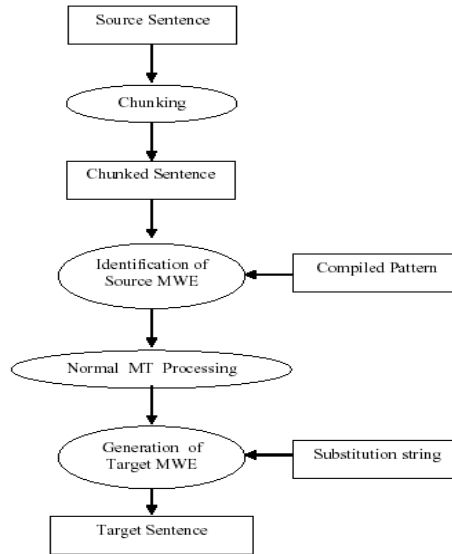


Fig. 3. Summary of processing an Actual Sentence

The elegance of the solution is that linguistic processing proceeds in the usual way as all other steps, giving great power to this approach. Learning pathway templates comprise of templates as well as the special steps, which are different from the usual steps. In the example above, the procedure part of the pathway consists of two special processing steps interspersed with the usual processing. Thus, it consists of both declarative as well as procedural parts where procedural parts consist of the special steps. As mentioned earlier, it connects language data (or patterns) to linguistic theory and the linguistic processing system.

5 Experiment

As the patterns that have already been analyzed manually by Linguists are available (SAID idioms dictionary, LDC), we conducted an experiment to evaluate the system based on that data. We picked a representative sample of 100 patterns along with their translations. The analyzed data was processed using the 2nd approach for compiling the pattern. The system performance was then evaluated.

6 Evaluation

The system is tested on a set of 230 sentences extracted from BNC corpus containing various MWEs, and the result is compared with output obtained without the specialized processing of MWEs.

Number of distinct MWEs	100
Number of sentences	230
Number of sentences in which translation of MWEs improved	139
Number of sentences in which translation of MWEs remained same	12
Number of sentences in which chunking was not compatible with the one required to process the MWEs	61
Number of sentences in which translation of MWEs was wrong	18

7 Major Limitations

1. Chunking output has errors. Hence, it does not match the compiled pattern and therefore, it is not processed.
For example, in the sentence,
“And striker Geoff Ferris is likely to put pen to paper for 12 months”
Here, the idiom “put pen to paper” was analysed by chunker as
((put)) [[pen]] ((to paper))
VG& NP VINF
Instead of
((put)) [[pen]] [[to paper]]
VG& NP PP
2. It has been observed that when ordinary linguists are asked to translate an idiom from one language to another (SAID idioms dictionary, LDC), they find it difficult to do it without looking at the example sentence containing the MWE. An example sentence helps the bilingual to deduce the meaning of an idiom in case of lack of familiarity. Hence, an example sentence is a must with every idiom in the dictionary.

8 Conclusion

In this paper, we have introduced a dynamic learning system that can take non-linguistic patterns for dealing with MWEs, interpret them linguistically, and use them in conjunction with the main linguistic system. This is made possible by the use of statistical taggers and specially designed learning pathway templates.

The system is carefully crafted so that it can be used by bilinguals to give patterns for handling MWEs. At the same time, it can be and it is implemented efficiently.

References

1. Abeille Anne and Schabes Yves, (1989). *Parsing idioms in lexicalized TAGs*. Proceedings of the 4th EACL, Manchester, UK.

2. Dekang Lin (1999). *Automatic identification of non-compositional phrases*. Proceedings of ACL'99, College Park, USA.
3. Eric Wehrli, (1998). *Translating idioms*. Proceedings of COLING ACL '98, Montreal, Canada.
4. Gael Dias, (2003). *Multiword Unit Hybrid Extraction*. Proceedings of the ACL-2003, Workshop on Multi-word Expressions: Analysis, Acquisition and Treatment.
5. Segond Frederique ; and Tapanainen Pasi, (1995). *Using a finite-state based formalism to identify and generate multiword expressions*. Technical Report MLTT-019, Rank Xerox Research Center, Grenoble, France.
6. Segond D., Giuseppe Valetto, E. Breidt (1996). *Formal Description of Multi-Word Lexemes with Finite-State Formalism IDAREX*. Proceedings of COLING '96.
7. Timothy Baldwin, Colin Bannard, Takaaki Tanaka and Dominic Widdows, (2003). *An Empirical Model of Multiword Expression Decomposability*. Proceedings of the ACL-2003, Workshop on Multiword Expressions: Analysis, Acquisition and Treatment.
8. Ralf D. Brown (1999). *Adding Linguistic Knowledge to a Lexical Example-Based Translation System*. Proceedings of the Eighth International Conference on Theoretical and Methodological issues in Machine translation.
9. Kenji Imamura, Eiichiro Sumita and Yuji Matsumoto (2003). *Feedback cleaning of Machine Translation Rules Using Automatic Evaluation*. Proceedings of 41st Annual Meeting of the Association for Computational Linguistics.