Extracting Multiword Expressions in Machine Translation from English to Urdu using Relational Data Approach

Kashif Bilal, Uzair Muhammad, Atif Khan, and M. Nasir Khan

***Abstract*—**Machine Translation, (hereafter in this document referred to as the "MT") faces a lot of complex problems from its origination. Extracting multiword expressions is also one of the complex problems in MT. Finding multiword expressions during translating a sentence from English into Urdu, through existing solutions, takes a lot of time and occupies system resources. We have designed a simple relational data approach, in which we simply set a bit in dictionary (database) for multiword, to find and handle multiword expression. This approach handles multiword efficiently.

***Keywords*—**Machine Translation, Multiword Expressions, Urdu language processing, POS (stands for Parts of Speech) Tagging for Urdu, Expert Systems.

I. Introduction

MULTIWORD expression is a term used to refer to such words which are combination of more than one words with same or different part of speech structure and giving single meaning. The main sources of multiword expressions are phrases and idioms. Other than idioms and phrases, other words also act as multiword expressions when combined together.

For example the word “foreign minister” is a multiword expression, which is formed by the combination of two different words “foreign” and “minister”.

However this word gives a single meaning. Similarly there is a good amount of multiword expression in English language corpora.

Resolving multiword expressions into their correct meaning in MT is a tedious job. Ivan A.Sag called it “a pain in the neck of NLP” [1]. Most of the translating systems use parsing module which contains a sub-module the tokenizer. The tokenizer makes token of the sentences. This is a mandatory task in MT. As a result, individual words are extracted and are input for the next process. It implies that if a sentence contains a multiword expression it losses this characteristic and thus having an adverse effect on the translation process.

MT specialists have designed many solutions to handle multiword expressions. In the subsequent discussion we will refer to some works of these solutions and finally we will present our approach that we used in AGHAZ (English to Urdu translator) to cop with the multiword expressions.

AGHAZ is translating software which translates tense-based English text into Urdu. It lies in the subcategory of Natural Language Processing (NLP) i.e. Machine Translation (MT). The translation is performed using dictionary and rule-based approach. It takes as input plain English text and generates their equivalent Urdu version provided the documents follow the grammar rules. The translated version of Urdu is in Unicode form. The standard followed for Unicode is ISO 8859-6, Unicode Standard 4.0, 1991-2003.

II. Various Approaches to Deal Multiword Expressions

As it mentioned earlier, the resolving multiword expressions is a tedious job. The reason is multiword expressions have different flavors. Various techniques have been presented to resolve this issue. Research papers like Ivan A.Sag’s “Multiword Expressions: a pain in the neck of NLP”[1], Scott S.L.Piao’s “Extracting Multiword Expressions with A Semantic Tagger” [2], Ann Copestake’s “Multiword Expressions: Linguistics precision and reusability” [3] provide some good solution to handle multiword expressions.

A very scarce amount of research has been done in MT from English to Urdu. To handle multiword expressions from English to Urdu Z. Pervez, S. Khan, F. Mustafa, M. Mahmood, U. Hasan have provided a ruled based solution, which they have adopted in their expert system MUTRAJUM, [4].

According to their solution when a sentence is resolved for multiword it applies the predefined rules by taking two words at a time. Then another next word is checked and rule is applied on it and so on. It implies that the procedure is highly recursive in nature and if a sentence contains some reasonable amount of multiwords, the frequency of recursion also increases. If we consider the best case to find multiword using

Kashif Bilal is Lecturer at COMSATS Institute of Information Technology Abbottabad, NWFP, Pakistan (e-mail: kas_atd1@yahoo.com; phone: 0092 300 5613174).

Uzair Muhammad is student at COMSATS Institute of Information Technology Wah Cantt, Pakistan (e-mail: joinuzair@yahoo.com; phone: 0092 300 9087797).

Atif Khan is a student at COMSATS Institute of Information Technology Wah Cantt, Pakistan (e-mail: atif_ciit@yahoo.com; phone: 0092-300 5051373).

M. Nasir Khan is a student at the COMSATS Institute of Information Technology Wah Cantt, Pakistan (e-mail: m_nasir_khan@yahoo.com; phone: 0092- 943 -412885).

recursion technique then it means that if there are 10 words in a sentence then almost 9 times you have to go through knowledge base, even if there is no multiword exists. This method is very costly. As MT is a notoriously slow process due to backtracking, dictionary lookup, rule application and words alignment the occurrence of recursion, could have negative impact on the speed of the process.

III. Aghaz Multiword Expression Handling

To resolve multiword expressions, we have used relational data approach. We have maintained a common dictionary to store words. It get a token and check whether its multiword bit is set or not, in case if it is set then it reserves that token and read the next token by look ahead. The multiword or phrase is then checked in the multiword knowledge base. In case it finds a solution there then it is retrieved while in case if there is no solution then it is backtracked and meaning for the first token is obtained from knowledge base and the next token is considered as a separate entity and then the same process is performed. Multiword bit is checked each time for every word except helping verbs, 'Not', and proper nouns. The key power is there if the bit is unset then it never goes to multiword knowledge base rather it picks the next token and hence a great amount of efficiency is achieved.
For example:

Consider the following sentence.

Shaukat Aziz is the prime minister of Pakistan.

[sentence1]

Prime minister is a multiword. According to algorithm "*Shaukat Aziz*" is never considered as multiword because this is Proper Noun.

It starts by scanning each token from left to right, and check its multiword bit. When *prime* is encountered it finds the multiword bit set. Now it get the next token which is *minister* and move to multiword knowledge base where it finds a solution.

IV. Conclusion

The technique adopted in AGHAZ to solve the multiword problem when implemented properly, can prove useful for any kind of multiword expressions, as no extra lookups to dictionary (database) is involved. The algorithm also has no immediate recursion. However there exists a little backtracking when we find a word with multiword tag set, i.e., to jump to multiword table. But results have shown it has a negligible effect on the processing speed of the translation.

References

[1] I.A. Sag, T. Baldwin, F. Bond, A. Copestake, D. Flickinger. 2001, "*Multi-word Expressions: A Pain in the Neck for NLP*", LinGO Working Paper No. 2001-03. Stanford University, CA.

[2] Scott S. L. Piao, Paul Rayson, Dawn Archer, Andrew Wilson, Tony McEnery, "*Extracting Multiword Expression Using a Semantic Tagger*", Lancaster University.

[3] Ann Copestake, Fabre Lambeau, Aline Villavicecio, Francis Bond, Timothy Baldwin, Ivan A.Sag, Dan Flickinger, "*Multiword Expressions: linguistic precision and reusability*", University of Cambridge Computer Laboratory,William Gates Building, JJ Thomson Avenue, Cambridge, CB3 0FD, UK. NTT Communication Science Labortries, Hikari Dai, Seiko-cho,Soraku-gun, Kioto 619-0237, JAPAN.

[4] Z. Pervez, S. Khan, F. Mustafa, M. Mahmood, U. Hasan, "*Pharasal Consolidation Algorithm For Part Of Speech Tags In Machine Translation From English To Urdu*", National University of Science and Technology, Rawalpindi Pakistan.

[5] Sarmad Hussein. "*Letter-to-Sound Conversion for Urdu Text-to-Speech System*". Center for Research in Urdu Language Processing, National University of Computer and Emerging Sciences, Lahore, Pakistan.

[6] T. Rahman (2002). "*Language Ideology and Power: Language Learning Among the Muslims of Pakistan and North India*", Oxford University Press, Karachi, Pakistan.

[7] Ethnologue, 13th Edition.

[8] T. Mitamura, E. Nyberg, E. Torrejon, D. Svoboda, A.Brunner and K. Baker. "Pronominal Anaphora Resolution in the Kantoo Multilingual Machine Translation System", Proceedings of the 9th International Conference on Theoretical and Methodological Issues in Machine Translation. Keihanna, Japan, Mar 2002.

[9] AltaVista Babelfish. URL: http://babelfish.altavista.com

[10] Google Language Tool.
URL: http://www.google.com.pk/language_tools

[11] Z. Pervez, S. Khan, F. Mustafa, M. Mahmood, U. Hasan, "*Pharasal Consolidation Algorithm for Part Of Speech Tags In Machine Translation from English to Urdu*", NUST Institute of Information Technology, National University of Sciences and Technology.

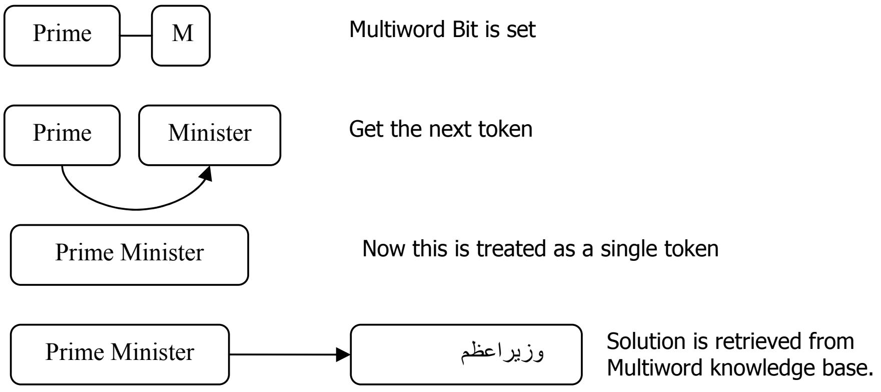

Fig. 1 Multiword recognition and Handling

Now consider another example. “Take” is a verb, but it creates lot of multiword i.e. take away, take off, take over etc.
We take tea.

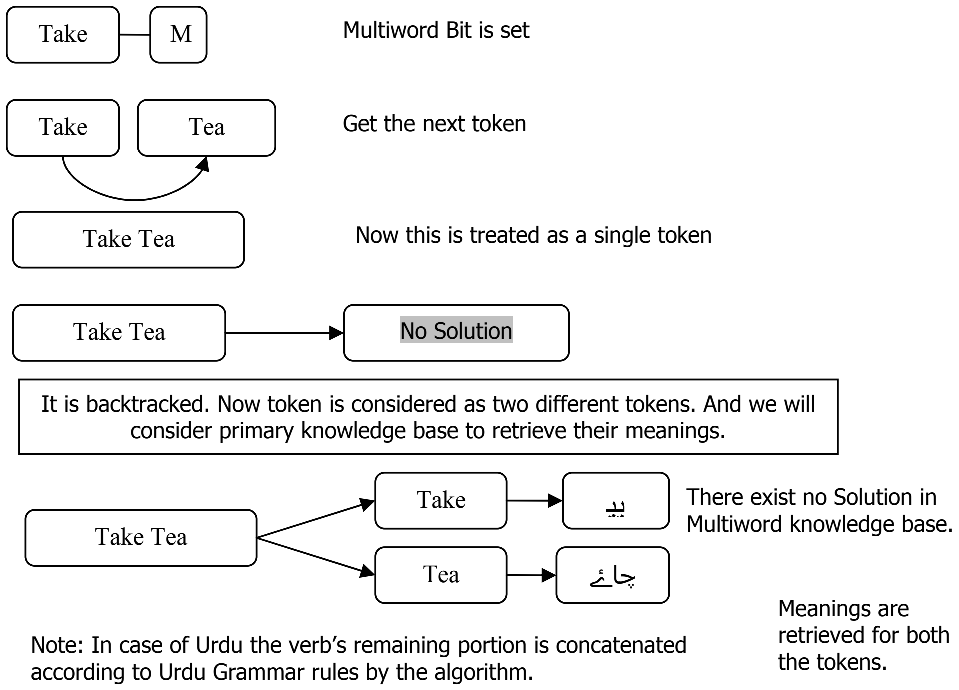
Take
M
Multiword Bit is set
Take
Tea
Get the next token
Take Tea
Now this is treated as a single token
Take Tea
No Solution
It is backtracked. Now token is considered as two different tokens. And we will consider primary knowledge base to retrieve their meanings.
Take Tea
Take
پیـ
There exist no Solution in Multiword knowledge base.
Tea
چائے
Meanings are retrieved for both the tokens.
Note: In case of Urdu the verb’s remaining portion is concatenated according to Urdu Grammar rules by the algorithm.

Fig. 2 Ambigous Multiword term and back tracking