Enhancing Translation Memories with Semantic Knowledge

The role of Translation Memories (TM) constantly grows in Translation Studies. This is the most popular application of Machine Translation[1]. In this paper we present an automatic method for building TMs enhanced with semantic knowledge. The majority of commercial TMs do not use linguistic information, but in (Gabor [5]) we find the description of a totally language aware TM, which aims at a better recall and precision than the existing systems. We propose template-driven approach enhanced with semantic features. Concepts from domain relevant ontology serve as semantic features, and the lexicon is mapped on the same ontology to solve several problems (e.g. ambiguity). Our first results of template extraction are presented and analyzed.

A template is a sequence of tokens and variables. Different formats of templates that use variables were found in literature: (Kaji [2]), (Güvenir and Cicekli [1]), (McTait [3]). Variables are inserted using different approaches: (Kaji [2]) uses a bilingual dictionary, parsers, and thesaurus; (Güvenir and Cicekli [1]) a lexical level representation; (McTait [3]) first uses a language-neutral technique based on the principle of string co-occurrence, and frequency threshold, followed by insertion of linguistic knowledge provided by POS tagger and morphological analysis.

There are two major problems solved by including semantics in the template format: learning false templates and disambiguation.

Considering the example English sentences, and their translations into Romanian,

1. He *marked* a paper/En -> El *a notat* o lucrare/Ro,
2. Stevenson *marked* a goal/En -> Stevenson *a marcat* un gol/Ro.

it can be noticed that the English verb *to mark* has two translations into Romanian: *a nota / a marca*, and because of this, a wrong template could be learned **X *marked* a Y**, where **X** (He/Pronoun, Stevenson/Proper Noun), **Y** (paper/Noun, goal/Noun).

For a new input sentence (e.g. The teacher *marked a* test), the right translation of the sequence of tokens in this example depends on the semantic value of the word contained in variable **Y** (here: "*test*").

Our approach extends the one described in (McTait [3]), by including semantic information into the template. We first apply a language neutral approach, based on surface forms as follows: we consider the tokens that occur in a minimum of ***n*** (frequency threshold) sentences form the sequence of tokens of the template; for tokens that differ in these ***n*** examples their semantic tags are compared, if they are the same – a semantic template is learned. In a pre-extraction phase, our corpus is semantically annotated with concepts from an ontology. Ontology granularity depends on the type of corpus, its domain, and availability of an already developed ontology for this certain domain. Generally speaking, we aim at using semantic features only were necessary. This way, the extracted templates are enriched with concepts, which we consider to be a good solution to the problems of learning false templates, and of ambiguity.

When we enrich our template with semantic information (ontology concepts, in our case) the templates look as follows:

1. **X** (person) *marked* **Y** (education),
2. **X** (person) *marked* **Y** (sport).

When the input is generalized: **X** (person) *marked* **Y** (education), it would be mapped on the first template. This is insured by the ontology containing the concepts necessary for disambiguation. Additionally the system uses a bilingual lexicon where for each entry (word), the concept in the ontology it refers to is specified (e.g. English / Romanian words – concept: paper / lucrare – education, paper / hartie – office, goal / gol – sport, test / test – education). As *mark* + an instance of (education) -> *a nota* and *mark* + an instance of (sport) -> *a marca*, the right translation for the input sentence mentioned before is produced.

1 More details on MT can be found in (Somers [4]).

Another problem solved by including semantic information into templates is disambiguation. This is the case of homonyms. Let us consider the following example:

The English word *paper* has at least two translations in Romanian: *hartie* (paper for writing) and *lucrare* (a test paper). If in the lexicon we do not have the concept specification, the translation might be wrong.

Based on the method described above we build the sequences of tokens that form the templates. We wanted to see what happens when we modify the frequency threshold.

We had as input an English text (news) of 56 sentences (478 tokens) and as output a list of all words and sequences of tokens that form the template (expressions). So far, no restrictions were considered (e.g. position of the tokens in a sentence, stop-list etc.)

The results of our experiments were the following:

Experiment	***Frequency threshold***	***New formed expressions***				***Comments***
		Total	***Perfect (XY)***	***Good (X ... Y)***	***Bad***	
1	2	112	19	26	78	-
2	3	40	6	15	29	there were lost 15 'perfect' expressions compared to Experiment 1. It was added 1 expression not present in Experiment 1.
3	4	17	3	1	13	There were lost 3 'perfect' expressions compared to Experiment 2 and 16 compared to Experiment 1.

Table 1. Experiment Results

In the above table, *'perfect*' are strings of **XY** form (token **Y** immediately follows **X**), **'***good*'- of **X ... Y** form (there are several tokens between **X** and **Y**) and '*bad*' are strings that do not form a template (e.g preposition + conjunction).

As it can be seen from Table 1., the first experiment gives the most '*perfect*' and '*good*' results. But also the most numerous '*bad*' results. In order not to lose '*perfect*' and '*good*' results, but to decrease the number of '*bad*' results there some constraints in the creation of the sequence of tokens should be introduced. One of the constraints can be the word order in the sentence.

In this paper we focused on the template extraction mechanism, which is based on the principle of string co-occurrence and frequency threshold, and analyzed some of the experiments we made. Our immediate future work includes improving template extraction results by considering position of tokens in a sentence, and the use of a stop list.

Some references:

[1] Güvenir H., Cicekli I.: 1998, "**Learning translation templates from examples**" in Information Systems 23, pp. 353-363.

[2] Kaji H., Kida Y., Morimoto Y.: 1992, "**Learning translation templates from bilingual text**" in Coling, pp. 672-678

[3] McTait K.: 2001, "**Linguistic Knowledge and Complexity in an EBMT System Based on Translation Patterns**", in the Proceedings of the "Example-Based Machine Translation" Workshop, MT-Summit VIII, Santiago de Compostela, Spain, September 18-22.

[4] Somers H.: 2000, "**Machine Translation**", in R. Dale, H. Moisl, H. Somers (eds), Handbook of Natural Language Processing, pp. 329-346.

[5] Gabor H., Tamas G., Balazs K.: 2004, "**Translation Memory as a Robust Example-based Translation System**", in EAMT, pp. 82-89.