動詞訳語選択のための　格フレーム木」の統計的な学習

田 中　　英 輝†

機械翻訳システムでは動詞の訳語を選択するために格フレームがよく利用される．格フレームは従来主として人手で記述されていたが，一貫性を保って記述するのが難しいこと，格フレームを部分的に変更した場合に起こる影響が把握しにくいことなどの重大な問題があった．そこでこれらの問題を解決するため，本論文では格フレームを決定木の形で表し (これを格フレーム木と呼ぶ)，これを英日の対訳コーパスから統計的な帰納学習プログラムを利用して学習することを提案する．本論文ではまず，この提案によって上記の問題が軽減される根拠を述べた後，本論文で作成した英日対訳コーパスについて述べる．続いて 7 つの英語動詞について格フレーム木の獲得実験を 2 つ報告する．最初の実験は，格要素の制約として英語の単語を使う格フレーム木を学習したものである．これにより得られた格フレーム木を観察したところ，人間の直観に近く，かつ直観を越えた非常に精密な訳し分けの情報が得られたことが明らかになった．次に，この格フレームの一般性を高めるために，英語の単語の代わりに意味分類コードを制約として利用する手法を提案し，これに基づいて格フレーム木を学習する実験を行った．得られた格フレーム木で未学習のデータの動詞の訳語を決定する評価を行ったところ，2.4%ないし 32.2%の誤訳率が達成された．この誤訳率と，先の英語単語を利用した格フレーム木での誤訳率との差は 13.6%ないし 55.3%となり，意味分類コードが有効に機能したことが示された．

キーワード:　機械翻訳，機械学習，訳語選択，コーパス，決定木，統計

Statistical Learning of "Case Frame Tree" for Translating English Verbs

Hideki Tanaka †

Our English to Japanese machine translation system uses surface verbal case frames (case frames) to select a Japanese translation for an English verb. The need to acquire and accumulate case frames leads directly to two problems:

- How to obtain transparency in the case frames? Case frames are sometimes changed after they are written and it is hard to predict how the translation selection is affected by these changes.
- How to keep consistency in the case elements and their restrictions? These elements should be used consistently because the matching calculation between case frames and syntactic structures (parser output) expects fair use of these elements.

To solve these problems, we propose two methods.

- Use of decision tree for case frame representation (case frame tree).
- Use of a statistical inductive learning algorithm to derive a case frame tree from a bilingual corpus.

The first method solves problem one: a change at any node in a case frame tree will affect only the translations under the node which is changed. The second method

solves problem two: the case elements and their restrictions are evaluated on the same basis according to their ability to distinguish the verb translations in the corpus. We used the learning algorithm C4.5, devised by Quinlan. C4.5 takes as input a table listing attribute, value, and class. To acquire a case frame tree, we replace attributes with case categories, values with restrictions of the case categories, and classes with Japanese translations of English verbs. We termed such a table a *Primitive Case Frame Table (PCFT)*. Before doing acquisition experiments on seven English verbs ("come", "get", "give", "go", "make", "run", "take"), we constructed an English and Japanese bilingual corpus from the AP (Associated Press) wirenews texts, a corpus that turned out to be about 6,000 translation pairs with syntactic tags. In the first experiment, we converted the corpus into the PCFT using all case categories appearing in the corpus and word forms for their restrictions. The acquired case frame trees basically duplicated the human work, but were far more precise in discriminating verb translations appearing in the corpus. Although the results indicate the basic effectiveness of our approach, the acquired case frame trees did not seem to have enough prediction power on open data since a lot of the word forms could be unknown words. To solve this problem, we generalized the word forms in the PCFT using semantic codes (Ruigo-Kokugo-Jiten, consisting of 4 digits) and then used C4.5. The five-fold cross-validation method was used to ensure the evaluation (error rate) precision. The error rate on open data for each verb was between 2.4% and 32.2%. Comparison of these figures with the baseline errors (error rates obtained by simply putting out the most frequent translation of a verb) showed a gain of between 13.6% to 55.3%, which indicates the basic effectiveness of using semantic codes. To lower error rate, we are devising an algorithm that can integrate word forms and semantic codes in an acquired case frame tree.

KeyWords: *Machine translation, Machine learning, Translation selection, Corpus, Decision tree, Statistics*

† NHK 放送技術研究所，先端制作技術研究部, NHK Science and Technical Research Laboratories, Program Production Technology Research Division