AMTA-2006

Panel: **Hybrid Machine Translation: Why and How?**

PANEL MODERATOR(S):
Violetta Cavalli-Sforza violetta@cs.cmu.edu
Alon Lavie alavie@cs.cmu.edu

PANELISTS:
Jaime Carbonell (CMU) jgc@cs.cmu.edu
Nizar Habash (Columbia U.) habash@cs.columbia.edu
Philipp Koehn (U. of Edinburgh) pkoehn@inf.ed.ac.uk
Stephanie Seneff (MIT) seneff@csail.mit.edu
John White (Systran) white@systransoft.com

In recent years, statistical machine translation has made great strides, example-based approaches have pushed forward, but systems based on linguistic knowledge have not been abandoned. In fact, we have seen the pendulum swing back a little towards the latter through the inclusion of linguistic knowledge in statistical systems and vice versa. Much current research in machine translation is neither based purely on linguistic knowledge nor on statistics, but includes some degree of hybridization. At AMTA 2004 and MT Summit 2005 just about all commercial MT developers also claimed to have hybrid systems. But is this mostly a good way to allow painting oneself into whatever paradigm that current "fashion" suggests one should be? And, given that no system still has approached human first draft quality, is hybridization little more than rearranging the furniture, or is there real progress or promise behind the mixing of different paradigms? Looking at it in the framework of the Vauquois' triangle, if the original SMT was just an automated return to the "direct" MT paradigm (the bottom of the triangle), are we now just ascending the same mountain again, but with different mountaineering tools? Will any hybrid of current paradigms take us any higher, more robustly, than in the past?

This panel/round table aims to explore the motivation for hybrid systems, the challenges they pose, and the benefits they offer, by addressing questions such as:

* Are there circumstances in which one of the traditional approaches (statistical, example-based, interlingua, transfer-based) is clearly better used alone than any of the others approaches used alone or in combination?

* In general, no MT developer strictly adheres to the claimed theoretical framework because there are many compromises in building a working system. Which components of a hybrid system tend to be empirically based and which are based on linguistic knowledge? In answering this question we can use the traditional subdivision into morphology, syntax and semantics, but are there other frameworks that we can apply?

* Some systems do use linguistic knowledge (morphology, grammar rules) but learn it from data through machine learning techniques. How successfully do they learn and to

what extent does the automatically learned knowledge contribute to overall system performance? Is automatically learned knowledge used together with hand-written rules and, if so, in what combination? Learning from data usually requires tagged training data of some sort, but the development of tagged corpora is not expense-free. Considering the cost of developing correctly tagged data, are automatic learning approaches really financially advantageous?

* What are the factors involved in determining which component(s) of a system to make statistical and which linguistically based? The scarcity of parallel corpora might push towards the choice of a rule-based approach, but where there is little parallel corpora, there is often also very limited funding or "commercial feasibility", making a rule-based approach too expensive. In other cases, the parallel corpora may exist in the hands of commercial translation companies or their clients, but may be expensive or impossible to obtain. For a particular application, given the choice of spending money to build linguistic knowledge or to buy or build a parallel corpus, which should be chosen? And how does one build a parallel corpus increase the chances of a maximally performing empirically-based system?

* Finally, hybrid systems may be providing performance gains and reducing some development expenses, but is this at the cost of more complex systems that are harder to document, understand, and use? And is the complexity making it increasingly difficult to perform blame/credit attribution and to ultimately further improve the overall performance of the system?