Application of Analogical Modelling to Example Based Machine Translation

Christos Malavazos[1, 2] **Stelios Piperidis**[1,2]
[1]Institute for Language and Speech Processing, [2]National Technical University of Athens
6 Artemidos & Epidavrou, 151 25 Marousi, Athens, Greece
{christos, spip}@ilsp.gr

Abstract

This paper describes a self-modelling, incremental algorithm for learning translation rules from existing bilingual corpora. The notions of supracontext and subcontext are extended to encompass bilingual information through simultaneous analogy on both source and target sentences and juxtaposition of corresponding results. Analogical modelling is performed during the learning phase and translation patterns are projected in a multi-dimensional analogical network. The proposed framework was evaluated on a small training corpus providing promising results. Suggestions to improve system performance are

1. Introduction

Ideally, an EBMT system must determine correspondences at a sub-sentence level if optimal adaptation of matching fragments is to be achieved (Collins, B., & Cunningham, P. 1995). In practice, EBMT systems that operate at sub-sentence level involve the dynamic derivation of the optimum length of segments of the input sentence by analysing the available parallel corpora. This requires a procedure for determining the best "cover" of an input text by segments of sentences contained in the database (Nirenburg, S. Domashnev, C., Grannes, D. 1993), (Cranias, L. et al 1994), (Frederking, R., Nirenburg, S., 1994), (Sato, S. 1995). What is needed is a procedure for aligning parallel texts at sub-sentence level, (Sadler, V., Vendelmans, R. 1990), (Boutsis, S., Piperidis, S. 1998). If sub-sentence alignment is available, the approach is fully automated but is quite vulnerable to the problem of low quality, as well as to translational ambiguity problems when the produced segments are rather small.

Several approaches aim at proceeding a step further, by attempting to build a transfer-rule base in the form of abstract representations through different types of generalization processes applied on the available corpora relying on different levels of linguistic information and processing (Kaji et al. 92), (Juola, P. 1994), (Furuse, O., Iida, H. 1996), (Veale, T. and Way, A. 1997), (McTait, K., et al. 1999), thus providing more complete "context" information to the translation phase. The deeper the linguistic analysis involved in such a process, the more flexible the final translation structures will be and the better the quality of the results. However, this kind of analysis unquestionably leads to more computationally expensive and difficult to obtain systems. Our approach consists in a fully modular analogical framework, which can cope with lack of resources, and will perform even better when these are available.

Analogical Modelling (AM) has been proposed as an alternative model of language usage. The main assumption underlying this approach is that many aspects of speaker performance are better accounted for in terms of "analogy", i.e. the identification of similarities and differences with forms in memory (the lexicon), than by referring to explicit and inaccessible rules. By "analogy" we mean the process of matching between an input pattern and a database of stored examples (exemplars). The result of this matching process is a collection of examples called the "analogical set" and classification of the input pattern is achieved through extrapolation from this set. At any given time, the main source of knowledge consists in a database of stored translation examples. These examples themselves are used to classify new items, without intermediate abstraction in the form of rules. In order to achieve this exhaustive database search is needed, and during this search, less relevant examples need to be discarded. All text features are equally important initially, and serve to partition the database into several disjoint classes of examples.

In contrast to most of the analogy-based systems our approach applies the same principles during the learning phase in an attempt to extract appropriate generalizations (translation rules) based on similarities and differences between input exemplars. In this way, analogy is treated as more

than simple pairwise similarity between input and database exemplars, rather it is considered as the main relation underlying a more complex network of relations between database exemplars.

2. General

The main idea behind our approach is based on the observation that given any source and target language sentence pair, any alteration of the source sentence will most likely result in one or more changes in the respective target, while it is also highly likely that constant and variable units of the source sentence correspond to constant and variable target units respectively. Apart from cases of so called "translational divergences" (Dorr, B. 1994) as well as cases of idiomatic expressions, in most cases the above assumption holds true. Especially in the case of technical sublanguages, where rather literal and accurate translation is expected, "translational divergences" are limited while idiomatic expressions can be captured and finally rejected from the main process, through certain constraints, as this will be explained later on.

The matching process as this is described by (Daelemans W., et al, 1997) based on Skousen's analogical modelling algorithm (Skousen, R. 1989), consists of two subsequent stages. The first stage of the matching process is the construction of "**subcontexts**", these are sets of examples and they are obtained by matching the input pattern, feature by feature, to each database item on an equal /not-equal base, and classify the database examples accordingly. Taking the input pattern ABC as an example eight ($=2^3$) different and mutually disjoint subcontexts would be constructed:

ABC, $\bar{A}$BC, A$\bar{B}$C, AB$\bar{C}$, $\bar{A}\bar{B}$C, $\bar{A}$B$\bar{C}$, A$\bar{B}\bar{C}$, $\bar{A}\bar{B}\bar{C}$

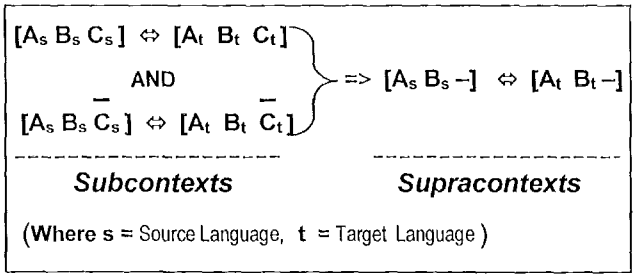
where the macron denotes complementation. Thus exemplars in the second class share only the second and third feature with the input pattern.

In the following stage "**supracontexts**" are constructed by generalising over specific feature values. This is done by systematically discarding features from the input pattern and taking the union of the subcontexts that are subsumed by this new pattern. Supracontexts can be ordered with respect to generality, so that most specific supracontext contains items that share all features with the input pattern while the less specific ones those items that share at least one feature. The most general supracontext contains all database examples whether or not they share any features with the input pattern.

Some exemplary **supracontexts** together with the respective **subcontexts** for the input pattern ***ABC*** are presented in the following table.

Supracontext	Subcontexts
A B –	ABC, AB$\bar{C}$
A – C	ABC, A$\bar{B}$C
– B C	ABC, $\bar{A}$BC
A – –	ABC, A$\bar{B}$C, AB$\bar{C}$, A$\bar{B}\bar{C}$

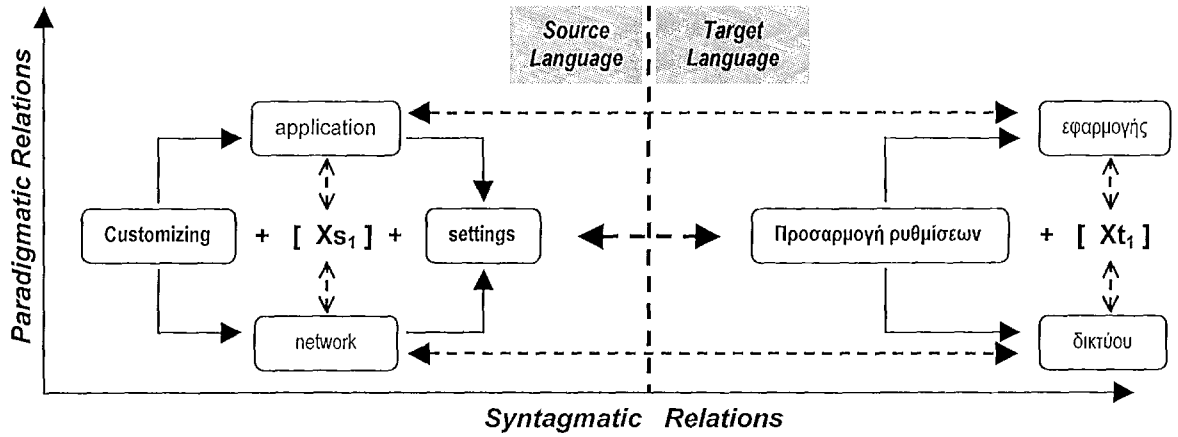
In addition, our approach introduces a second dimension to the above described process, that of language, by simultaneously performing the matching process to target language equivalents and aligning individual results, based on the principles described earlier. Therefore, what we are ultimately searching for, are source and target sentence pairs for which evidence of correspondence between any or all of respective subcontexts within the available training corpora is available. This will subsequently lead to links between respective supracontexts. For example :



3. The learning mechanism

3.1 Translation Templates

Supracontexts and translation templates can be viewed as two sides of the same coin. Generalization through unification on feature values of neighbouring sentences, if these satisfy certain criteria, leads to more abstract expressions of bilingual pairs of "pseudo-sentences", consisting of sequences of constant and variable elements,



(1): **Customizing** *application* **settings** ⇔ **Προσαρμογή ρυθμίσεων** *εφαρμογής*, and
(2): **Customizing** *network* **settings** ⇔ **Προσαρμογή ρυθμίσεων** *δικτύου* is :

(A): Customizing + [Xs_1] + settings ⇔ Προσαρμογή ρυθμίσεων + [Xt_1]

where : **[Xs_1] ⇔ [Xt_1] and,**

application ⇔ εφαρμογής
network ⇔ δικτύου

Figure 1

where variable elements are represented by special symbols ("X_i") and constant-fixed elements act as the context in each case.

3.2 Translation Units

Discarded features (represented by the "–" symbol) of corresponding supracontexts, rising from variable elements of the matching sentences, correspond to the translation units of the respective translation patterns. As a result, single or multi-word elements (translation units) of source and target language appearing within corresponding supracontext positions, are linked and stored, comprising the bilingual translation unit lexicon.

3.3 The Analogical Network

The main linguistic object for which matching is performed is not the sentence but pairs of source and target sentences/exemplars. Therefore, matching between linguistic objects is performed in two dimensions simultaneously, that is between source and target sentences of matching pairs respectively. The result of the process, if certain conditions are met, are stored in an "analogical network" (Federici, S. & Pirrelli V., 1994) of inter-sentence and intrasentence relations between these exemplars and their generalizations. A rather simple example of this is presented in Figure 1.

Different parts of matching sentences are replaced by corresponding **variables,** and are consequently assigned the role of **translation units**, while **similar/constant** parts are considered to be the **context** under which variable units are instantiated. The union of context and variables establishes the "generalized" translation (paradigmatic) patterns between source and target language. The similar (constant) and different (variable) parts between source and target sentences are factored out and presented as separate nodes in the above diagram.

For each sentence we can view its constituent single or multi-word, constant or variable units as separate nodes, where links between these nodes indicate the syntagmatic relations between them, that is, the way they actually appear and are ordered in the respective sentence. The vertical axis represents the paradigmatic dimension of available alternants, that is, the information concerning which substrings are in complementary distribution with respect to the same syntagmatic context i.e. with respect to the same context "Customizing __

settings". Syntagmatic links constitute the intrasentence relations/links between sentence constituents while paradigmatic ones correspond to the intersentential relations. Furthermore, a third dimension is added to the whole framework, that of the **"language"**, since all principles are applied simultaneously to both source sentences and their target equivalents. In case, linguistic annotations are available, they are appropriately incorporated in the respective nodes.

At this point no conflicts are resolved. All possible patterns are stored in the network including conflicting as well as overlapping patterns. However, all links both paradigmatic and syntagmatic are weighted by frequency information. This will eventually provide the necessary information to disable and even discard certain false or useless variables or templates.

3.4 The Algorithm

Translation templates as well as translation units are treated as paradigmatic flexible structures that depend on the available evidence. As new data come into the system, rules can be extended or even replaced by other more general ones. It is usually assumed that there is only one fixed way to assign a structural representation to a symbolic object either be a translation unit or a translation template. However, it is obvious that in our approach there is no initial fixed definition of this particular structure, rather it is left up to the training corpus and the learning mechanism. As was expected, under this kind of analogy-based approach, linguistic objects were determined based on the paradigmatic context they appeared in, resulting in a more flexible and also corpus dependent definition of translation units.

Search Space Reduction

In general, if sentence matching were unconstrained and all resulting matches were stored in the analogical network, then the number of all links (inter/intra-sentential) for N equal to the number of translation patterns learned through the process and L equal to the number of words in a sentence (template) would be :

$$O\left(\sum_{i=1}^{N}(i-1)\times 2^{L}\right)$$

while the complexity of the learning phase is also increased by the fact that each candidate rule needs to be verified against the available corpus, introducing an additional parameter **S**, that of the size of the training corpora (in number of sentences).

Moreover, if a rather straightforward approach in matching was to be followed, the complexity involved for each individual candidate sentence would be enormous. In such an approach, for each candidate sentence, all corresponding subcontexts would have to be identified and verified against the available corpora. For instance, a sentence of length **L** would generate $\mathbf{2^L}$ subcontexts, thus resulting in $\mathbf{O(2^L)}$ required search actions against the available corpora. Even if constraints would be set upon the length of possible ignore (variable) areas, for example = 5 words, the process would still be too complex. For example for a sentence of length **L = 10** and for variables of length up to **5** words, the possible subcontexts that have to be matched against the corpus would be $\binom{L}{1}+\binom{L}{2}+\binom{L}{3}+\binom{L}{4}+\binom{L}{5} = 10 + 45 + 120 + 210 + 36 = 421$, where terms of the previous equation correspond to the subcontexts with variables of length 1 to 5 respectively.

The SSR methodology, depends on the specific needs of the particular task. Run-time pruning of possible matches can speed up the learning process, however it also reduces system recall & coverage. On the other hand, constraints on paradigmatic relations are more reliable providing better results but cannot contribute to the speed of the learning process. SSR was based on an efficient indexing and retrieval mechanism (Willman, N. 1994) allowing fast identification of "relevant" sentences based on common single/multi-word units. In this way, the search space for each individual candidate was significantly reduced to a smaller set of possible matching sentences.

Distance Metric

The main objects of knowledge generated by the learning process are the translation patterns and the bilingual lexicon of translation units. During the learning process, both sources are enriched when possible. Sentences are analysed and encoded to two-dimensional vectors based on the words (first dimension) and the linguistic annotations (second dimension) they might contain. Then sentence vectors are compared on an equal - not equal basis

through a Levensthein or Edit distance algorithm (Damerau, F. 1964), (Oflazer, K. 1996). The algorithm, implemented through a dynamic programming framework (Stephen, G. 1992), computes the minimum number of required editing actions (insertions, deletions, substitutions, movements and transpositions) in order to transform one sentence into another through an inverse backtracking procedure. The final similarity score is computed by assigning appropriate weights to these actions. For the time being only insertions and deletions were accounted for. More complex actions, like transpositions or movements of words and their influence in the final translation pattern will be the focus of future work.

Variable Elements

Differences between matching sentences result in coupling of corresponding source and target words, as explained earlier in this section, thus enriching the lexicon with new information. Coupling is restricted to content words. Content words can usually be replaced by other words of the same category acting as potential variables (Kaji, H. et al 1992). On the other hand functional words do present an "abnormal" translational behavior, since they sometimes act as optional units which do not appear in both source and target segments, other times have a one-to-one correspondence, yet it is not rare that they affect the target pattern (especially when they participate in verb complementation). "Exclusion lists" were used for this purpose in order to reject functional words from acting as translation variables.

Workflow

All sentences are stored as vectors of constituent words-annotations. Functional words are marked as such. The process runs iteratively for all sentences starting from sentences of length 1 to the maximum length appearing in the training corpus. The process terminates in case of an unsuccessful loop, meaning an iteration where no new information either translation units or templates were extracted. The learning process consisting of five subsequent phases, is depicted in detail in Figure 2 :

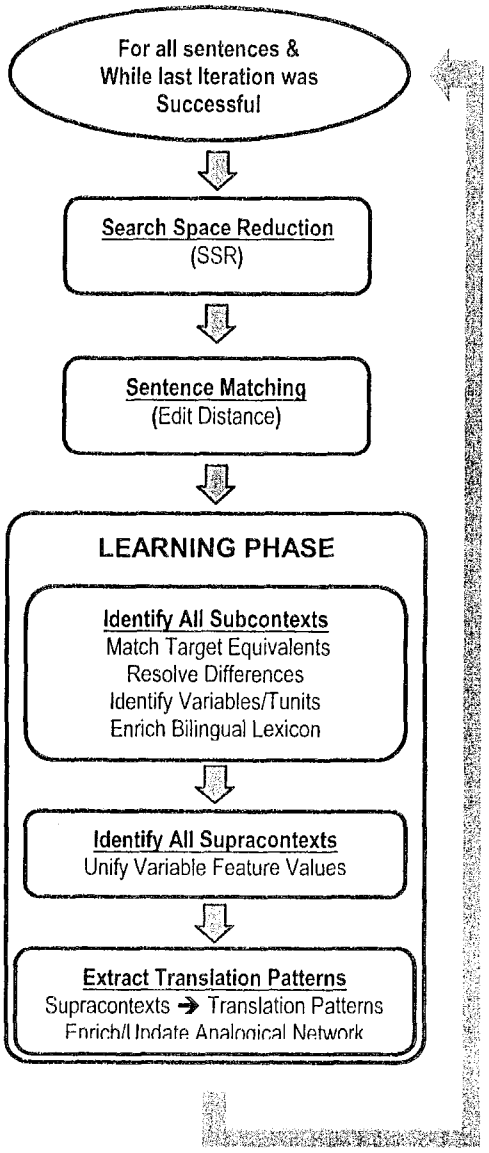


Figure. 2

Phase1 Search Space Reduction : Extract an initial set of possibly relevant sentences for the current input sentence.

Phase2 Sentence Matching : Match Input sentence against the previous set. Matching candidates are sorted based on distance score. Matches with fewer differences are examined first.

Phase3 Identification of Subcontexts : For each matching candidate, identify the respective subcontext of the input sentence that it adheres to. Examine target language equivalents. Resolve differences between source and target language matching candidates based on already existing information contained in the bilingual lexicon. During this process, the bilingual translation unit lexicon is enriched with any successfully resolved difference (even if the particular candidate will not finally lead to a new translation pattern).

Phase4 Identification of Supracontexts : Based on the already identified subcontexts produce the respective supracontexts through unification of respective variable feature values.

Phase5 Extraction of Translation Patterns : Construct corresponding translation patterns from existing supracontexts. Update analogical network. In case a pattern already exists, update the weight of its constituent links.

At the end of the learning process the analogical network has been enriched with all possible translation patterns and variables/units extracted from the available corpora. Conflict resolution and network refinement in general is performed on the final results, where all information is available as described in the next section.

3.5 Network Refinement

As mentioned earlier, the analogical network contains all translation alternatives for individual translation units as well as all translation patterns resulting from the learning process. However, link weight information is also included in the above framework representing the validity of a particular relation against the training corpus.

Translation alternatives of individual units (in our case words) are implicitly classified through their context, that is the constant part of the translation patterns they participate in. These will constitute the main selection criterion during translation. However, frequency information is also used in order to disable and finally discard obsolete or erroneous translation unit alternatives.

Translation templates are compared with respect to their source and target language constituent patterns: **(a) Conflicting templates**, that is templates sharing only one of the two patterns are subsequently checked in terms of weight information. Templates of equivalent weights are considered equally effective. This is usually the case where different translations are produced from the same source pattern due to semantic differences on the variables it contains. Conflicting templates with significantly low weights (under a predefined threshold), are judged ineffective or "exceptional" (Nomiyama, H. 1992) and are flagged as such in order to receive a special treatment during the translation phase (Watanabe, H. 1994). These can even be disabled or discarded from the network depending on their significance weight through a dynamic "forgetting and remembering" process (Streiter, O. et al, 1999). **(b) Overlapping templates**, where both source and target patterns of one template can be generated from the other by coupling words of the constant part of the template through valid translation alternatives included in the network, are identified and the more general ones are preferred. A basic requirement is that the set of all translation alternatives instantiating the variables of the more general template is a superset of those instantiating the less general one. In any other case, both templates are retained. And finally, **(c) complementary templates**, are also identified and replaced by their union.

4. Evaluation

The training set consisted of a bilingual (EN-GR) technical corpus (automotive industry) of 5K sentences, ~20K wordforms on each language. The process resulted in ~550 translation rules, and 350 translation units (~50 multi-word ones). The precision estimated through manual evaluation was ~75%. More than 23% of the erroneous rules were due to idiomatic expressions. The rest of the errors was caused by imprecise translation patterns found in the corpus. However, these errors being rather exceptional, received a very low weight of effectiveness at the end of the process. No straight forward approach to measure the recall of the learning process was devised, since it was not easy to a-priori determine the number of rules that should be extracted from the training corpus. However, coverage of the final translation rule set against the corpus was measured and found equal to 38%. More specifically, the set of 500 rules could through an inverse process generate 38% of the corpus sentences, subsequently interpreted in a significant gain in terms of storage space. Another obvious benefit is the subsentential alignment information that is, the source and target translation units learned at the end of the process.

5. Conclusion & Future Work

We have presented a self-modelling, incremental analogical algorithm for extracting translation patterns from existing bilingual corpora as well as a method for efficient storage and representation of extracted relations between various units of text. Not surprisingly, the quality of the results depends on the available information in terms of quantity as

well as quality and depth. Lack of any kind of linguistic information will consequently result in translation rules based only on "shallow" evidence. Similarly, information of low quality will generate erroneous rules. However, this is a basic presupposition of any EBMT system: "what you give is what you get...".

The proposed framework was initially evaluated based only on string form information. However, the model can easily take into account "deeper" linguistic knowledge during the learning phase, thus improving the quality of the final results. Evaluation of learning performance in this case is the main object of current work.

Another, interesting issue is how the current framework can constrain acceptable multi-word variables in order to reduce computational complexity. In present, accepting or rejecting candidate variables extracted from the sentence matching process, is based on a simple heuristic of length in content words. This type of approach would presumably require some kind of clue on what could be an acceptable translation unit pattern (Juola, P. 1994), (Furuse, O., Iida, H. 1996).

Finally, future work will mainly focus on how the system can invoke all existing information in order to generate new translations, mainly aiming at automatic and (semi-) automatic methods for "recursive" as well as "parallel" utilization of multiple translation rules towards optimal "coverage" of new incoming sentences.

7. References

(Boutsis, S., Piperidis, S. 1998) Aligning Clauses in Parallel Texts. *3rd Conference on Empirical Methods in Natural Language Processing, June 1998*

(Collins, B., & Cunningham, P. 1995) A Methodology for EBMT. *4th International Conference on the Cognitive Science of Natural Language Processing, Dublin 1995.*

(Cranias, L., Papageorgiou, H. and Piperidis, S. 1994). A matching technique in Example-Based Machine Translation, *Proc. of COLING-94, pp 100-105.*

(Daelemans, W., Gillis, S. & Durieux, G., 1997) Skousen's analogical modelling algorithm: a comparison with lazy learning. *New Methods in Language Processing: Edited by Daniel Jones & Harold Somers, UCL Press, p.3-15.*

(Damerau, F. 1964) A Technique for Computer Detection and Correction of Spelling Errors. *Communications of the ACM 7, p. 171-176, 1964.*

(Dorr, B. 1994) Machine Translation Divergences: A Formal Description and Proposed Solution. *Association for Computational Linguistics, Vol. 20, 1994.*

(Federici, S. & Pirrelli, V. 1994). The compilation of large pronunciation lexica: the elicitation of letter to sound patterns through analogy based networks. *Papers in Computational Lexicography, Complex '94, Budapest, 59-67.*

(Frederking, R., Nirenburg, S., 1994) Three Heads are Better then One. *Proceedings of the fourth Conference on Applied Natural Language Processing, ANLP-94, Stuttgart, Germany*

(Furuse, O., Iida, H. 1996) Incremental Translation Utilizing Constituent Boundary Patterns. *Proc. Coling-96, pp 412-417.*

(Juola, P. 1994) Self-Organizing Machine Translation: Example-Driven Induction of Transfer Functions. *University of Colorado at Boulder, Technical Report CU-CS-722-94.*

(Kaji, H., Kida, Y., and Morimoto, Y., 1992) Learning Translation Templates from Bilingual *Text. Proc. Coling., p. 672-678, 1992.*

(McTait, K., Olohan, M., Trujillo, A. 1999) A Building Blocks Approach to Translation Memory. *Proc. From the 21st ASLIB Conference, London, 1999.*

(Nirenburg, S. Domashnev, C., Grannes, D. 1993) Two Approaches to Matching in Example-Based Machine Translation. *Proc. of TMI-93, Kyoto, Japan, 1993.*

(Nomiyama, H. 1992) Machine Translation by Case Generalization. *Proceedings of the [sic] International Conference on Computational Linguistics, COLING-92, Nantes, p.714-720.*

(Oflazer, K. 1996) Error-tolerant Finite State Recognition with Applications to Morphological Analysis and Spelling Correction. *Association for Computational Linguistics, Vol. 22, (1), 1996*

(Sadler, V., Vendelmans, R. 1990) Pilot Implementation of a Bilingual Knowledge Bank. *Proc. of Coling, pp 449-451, 1990.*

(Sato, S. 1995). MBT2: A Method for Combining Fragments of Examples in Example-Based Machine Translation. *Artificial Intelligence 75, 31-49.*

(Skousen, R. 1989) Analogical Modelling of language. *Dordrecht: Kluwer.*

(Stephen, G. 1992) String Search. *University College of North Wales, Technical Report TR-92-gas-01.*

(Streiter,O., Iomdin,L., Hong,M., Hauck,U., 1999) *IAI CAT2 Publications, www.iai.uni-sb.de*

(Veale, T. and Way, A. 1997) Gaijin: A Bootstrapping Approach to Example-Based Machine Translation. *International Conf., Recent Advances in Natural Language Processing, Tzigov Chark, Bulgaria, 239-244.*

(Watanabe, H. 1994) A Method for Distinguishing Exceptional or General Examples in Example-Based Transfer Systems. *The 15th International Conference on Computational Linguistics, COLING-94, Kyoto, p.39-44.*

(Willman, N. 1994) A Prototype Information Retrieval System to Perform a Best-Match Search for Names. *Conference Proceeding of RIAO '94.*