A review of EBMT using proportional analogies

Harold Somers **Sandipan Dandapat** **Sudip Kumar Naskar**
Centre for Next Generation Localisation
Dublin City University
Glasnevin, Dublin 9, Ireland
{hsomers,sdandapat,snaskar}@computing.dcu.ie

Abstract

Some years ago a number of papers reported an experimental implementation of Example Based Machine Translation (EBMT) using Proportional Analogy. This approach, a type of analogical learning, was attractive because of its simplicity; and the papers reported considerable success with the method. This paper reviews what we believe to be the totality of research reported using this method, as an introduction to our own experiments in this framework, reported in a companion paper. We report first some lack of clarity in the previously published work, and then report our findings that the purity of the proportional analogy approach imposes huge run-time complexity for the EBMT task even when heuristics as hinted at in the original literature are applied to reduce the amount of computation.

1 Introduction

At the last workshop on EBMT four years ago, Lepage and Denoual [10] presented "the 'purest' EBMT system ever built: no variables, no templates, no training, examples, just examples, only examples". This purely data-driven approach uses the notion of **proportional analogies**, a type of analogical learning, the very simplicity of which is its attraction. In [12] the process is explained in considerably more detail. The approach, which was also presented in [11], was re-implemented with some modifications in [13], and with a change of programming platform in [14]. These five papers represent, as far as we can tell, the only full-scale implementations of EBMT using proportional analogies for the entire translation task. The idea is adapted to translating unknown words in the context of another approach to MT as reported by Langlais and colleagues in [2,4,6,7].

In our own work, we have been trying to address the problem of translating between English and Bangla (and other Indian languages), mainly in a statistical MT framework. Faced with mounting difficulties and low quality, we were inspired to revisit the pure EBMT approach. In this paper we critically review the literature on experiments in this framework, and make some observations on some details of the implementation of the proportional analogy approach, not always entirely specific in the existing literature, and draw some conclusions on the best scenario for use of this approach. We plan to report our own quite varied results in the near future.

2 The underlying idea

2.1 Proportional analogies

Proportional analogies are statements of the relationship between four entities as in (1),

(1) $A : B :: C : D$

read as "*A* is to *B* as *C* is to *D*". The '::' symbol is sometimes replaced with an equals sign, implying that statements such as (1) are effectively equations, susceptible to zero, one or more solutions if any of the entities (usually *D*) is taken as a variable. Noted long ago by the likes of Aristotle and Plato, proportional analogies are often seen as a way of expressing our knowledge of the world (2a), and the lexical relations that encode it (2b), and for this reason have been popular as a means of knowledge representation in Cognitive Psychology and Artificial Intelligence. In Linguistics, they are often used to explain historical language change, especially when previously unattested forms start to appear (2c) [1, p.104f].

Mikel L. Forcada, Andy Way (eds.)
Proceedings of the 3rd Workshop on Example-Based Machine Translation, p. 53–60
Dublin, Ireland, November 2009

(2) a. lungs are to humans as gills are to fish
b. *cat* : *kitten* :: *dog* : *puppy*
c. *speak* : *spoken* :: *break* : *broken*

Lepage had earlier [8] developed an algorithm that could solve analogical equations over strings of characters, based on finding the longest common subsequences, and measuring edit distance. Lepage showed with examples from various languages that his algorithm could handle insertion/deletion of prefixes and suffixes (3a), exchange of prefixes/suffixes (3b), infixing and umlaut (3c), and parallel infixing (3d).

(3) a. (French) *répression* : *répressionnaire* :: *réaction* : $\boldsymbol{x} \Rightarrow \boldsymbol{x}$ = *réactionnaire*
b. *wolf* : *wolves* :: *leaf* : $\boldsymbol{x} \Rightarrow \boldsymbol{x}$ = *leaves*
c. (German) *fliehen* : *floh* :: *schließen* : $\boldsymbol{x} \Rightarrow \boldsymbol{x}$ = *schloß*
d. (Proto-Semitic) *yasriqu* : *sariq* :: *yanqimu* : $\boldsymbol{x} \Rightarrow \boldsymbol{x}$ = *naqim*

2.2 Analogy-based EBMT

Using the algorithm from [8], Lepage and Denoual [10,11,12] show how an EBMT system can be built up. Treating sentences as strings of characters (including spaces), they note that proportional analogies can be handled, as in (4).

(4) *They swam in the sea* : *They swam across the river* :: *It floated in the sea* : *It floated across the river*

For the purpose of EBMT of course we must assume a database of example *pairs*, where each sentence has a corresponding translation. For the first three sentences in (4), these are given in (5).

(5) a. *Nadarón en el mar*.
b. *Atraversarón el río nadando*.
c. *Flotó en el mar*.

Suppose now that we want to translate the sentence *It floated across the river*. Informally, the translation process is as follows:

1. Find a triple of sentences in the example set that satisfy the analogical equation in (6).

(6) *A* : *B* :: *C* : *It floated across the river*

2. Take the translations corresponding to *A*, *B* and *C* (notated *A′*, *B′*, *C′*).

3. Solve the analogical equation in (7): $\boldsymbol{x}$ represents the desired translation.

(7) *A′* : *B′* :: *C′* : $\boldsymbol{x}$

Substituting the three sentences in (5) into (7), we have a solvable analogical equation with $\boldsymbol{x}$ = *Atraversaró el río flotando*, which is an acceptable translation.

The steps above represent the ideal performance. As described in [10, p.83; 12, p.259], steps 2 and 3 can take a different form however. In their description, step 1 involves first selecting relevant sentence *pairs* (*A*, *B*), where relevance w.r.t. *D* is defined by "a similarity criterion". Where *C* is found in the example database, step 2 proceeds as above. If *C* is not in the corpus, then the translation *C′* is arrived at "using the present method recursively". This recursion is briefly discussed in [10, p.88] and [12, p.278], where it is noted that one recursive call is needed on average, with a maximum of two, all this showing how close the sentences in the test set were to the resource used. This is an issue to which we will return below.

2.3 Some immediate difficulties

The simple process outlined above hides two potential difficulties, both noted by Lepage and Denoual, but with significant processing implications.

The first is that for a given input sentence, *D*, the database may contain multiple triples (*A*, *B*, *C*) that offer a solvable analogy, as shown in (8).

		A		*B*		*C*		*D*
(8)	a.	*They swam in the sea*	:	*They swam across the river*	::	*It floated in the sea*	:	*It floated across the river*
	b.	*It walks across the street*	:	*It walked across the street*	::	*It floats across the river*	:	*It floated across the river*
	c.	*He swam*	:	*He floated*	::	*It swam across the river*	:	*It floated across the river*

Furthermore, because of the unconstrained nature of proportional analogy as a mechanism, there is always the possibility of "false analogies", that is, sets of strings for which the analogy holds, but which do not represent a linguistic relationship. A good example is (9), where the *A*:*B* relationship is a simple one-character substitution (*p* for *a*), mirrored in the case of *C*:*D*.

(9) *Yea* : *Yep* :: *At five a.m.* : *At five p.m.*

Example (9) is taken from the Basic Traveler's Expression Corpus (BTEC), which Lepage and Denoual use in the experiments to be described below. Lepage estimated [9] that less than 4% of the analogies findable in the BTEC data would be of this kind [12, p.271].

The second problem, also due to the unconstrained nature of the mechanism, is that (7) may have multiple solutions. To take another of their examples, the solution to (10) could be any of the strings in (11). The equation requires us to substitute *May I have* with *I'd like … strong*, and *some tea please* with *a cup of coffee*, but nothing in the algorithm tells us where to insert the word *strong*, and, remembering that we are treating the sentences as strings of characters rather than strings of words, nothing prevents the string from being inserted as in (11d,e) etc.

(10)	*May I have some tea, please?*	:	*May I have a cup of coffee?*	::	*I'd like some strong tea please*	:	***x***

(11) a. *I'd like a strong cup of coffee.*
b. *I'd like a cup strong of coffee.*
c. *I'd like a cup of strong coffee.*
d. *I'd like a cstrongup of coffee.*
e. *I'd like a custrongp of coffee*
etc.

The proportional analogy method can consider the examples to be either strings of characters, or strings of words. The latter approach of course eliminates the possibility of outputs such as (11d,e), but also means that correspondences such as *walks* : *walked* :: *floats* : *floated* seen in (8b) above would not be captured.

3 Implementation

As mentioned above, the method involves two basic operations: searching for possible analogical equations, and solving them. The latter of these is relatively straightforward, and various algorithms have been proposed, Lepage's own implementation [8] being generally accepted as the most efficient.

More problematic is finding the triple of (*A*, *B*, *C*) that satisfy the analogical equation (1) for a given *D*. Noting that the algorithm is in principle cubic in the amount of data, (or "only" quadratic if one searches first for (*A*, *B*) pairs), commentators agree that the search is intractable or unmanageable, except for "toy problems". Even if the analogy solver is quite efficient, it is obvious that some heuristics are needed to reduce the search space.

3.1 Heuristics

Lepage and Denoual initially [10, p88; 11] report "using a simple heuristics [sic] to select only relevant pairs entering in analogical equations … to keep translation times reasonable". Only in [12] do they make explicit this heuristic:

H1: Consider as candidates only sentences whose length is more than half and less than double the length of the input sentence [12, p.259].

This is of course a risky strategy – it would discount a pair such as in (12) – and as they note [10, p.88], although an average of 689K equations are formed to translate one sentence when using a corpus of around 100K sentence pairs, only 28% of analogical equations are solved. "Future work should include finding a heuristic[] that would increase this proportion so as to reduce the number of unnecessary trials."

(12) a. *He dived.* [9 characters]
b. *He dived into the river.* [24 characters]

The implementation described in [13] introduces two heuristics to speed up the search for (*A*,*B*) pairs:[1] the corpus is sorted relative to the sentence to be translated (*D*), using edit distance for the selection of *A*s, and, in what can be seen as an adaptation of H1, selection of *B*s by "inclusion score", i.e. length of *B* minus its similarity to *D*.

H2: Consider as candidates primarily sentence pairs where *A* has a low edit distance w.r.t. *D*, and *B* has a low inclusion score.

Our use of the word "primarily" here is intended to reflect the idea that the search space should be ordered, and the most likely solutions tried first, the implication being that the number of solutions sought will be limited (possibly to just the first one found), so there is a premium on trying the most promising paths first.

Langlais and Patry [4] similarly reduce the search space by requiring both *A*s and *B*s to be within a certain edit distance of *D*, though they do not specify what that distance is in their experiments. Langlais and Yvon [6] employ a similar heuristic as one of several, which they try to evaluate separately (called ED, see below).

Lepage and Lardilleux [13] suggest a further heuristic, also involving sentence pairs which are close, and/or where one subsumes another, but their explanation is unclear. Lepage et al. [14] introduce a heuristic, which may be the same as the badly explained one in [13], based on the observation that "strings which exhibit smaller prefix or suffix differences relative to the input string [*D*] seem better candidates for [*A*]" [14, p.43]. This heuristic involves taking pairs in order of longest common substring (LCS – by which is meant longest contiguous substring, which is not the same as the more familiar longest common subsequence). As a further measure, the data is searched for substrings limited to length $|S|-1$, $|S|-2$ for an LCS of length $|S|$, and (separately) for *n*-grams with $n = 1,2,3$. Note that in this implementation the basic unit is a word rather than a character. Langlais and Yvon [6] employ a similar shared *n*-gram-based heuristic, though theirs are character *n*-grams.

H3: Consider as candidates primarily sentence pairs where *A* and *B* have the largest LCS (or signifcant *n*-grams) in common with *D*, starting with *A*s and *B*s that share the longest LCS (or significant *n*-grams) with each other, and with LCSs of a similar length.

Another heuristic or "trick" in [6], called S-TRICK, relies on the property expressed in (13).

(13) $[A : B :: C : D] \Rightarrow$
$A[1] \in \{B[1],C[1]\} \vee D[1] \in \{B[1],C[1]\}$
$A[\$] \in \{B[\$],C[\$]\} \vee D[\$] \in \{B[\$],C[\$]\}$
where $S[1]$ and $S[\$\}$ are the first and last symbols , respectively, in the string *S*.

The trick is to limit the search to triples that pass this test.

H4: Consider as candidates only pairs where *B* or *C* share the same first or last symbol with *A* or *D*.

A final reported heuristic relates to the effort in solving target-side analogical equations $A' : B' :: C' : \boldsymbol{x}$, which almost all authors report as being potentially time consuming, with only a small

[1] In the reports reviewed here, authors express the basic analogical equation (1) in a variety of ways, in some cases still using *A*,*B*,*C*,*D* but arranged differently (e.g describing the task as attempting to translate *A* by solving the equation $A : x :: C : D$), or using *x*,*y*,*z*,*t*. In this discussion we recast all formulae to follow our equation (1), and sometimes therefore have to rephrase cited examples. So *D* is always the input sentence, and $\boldsymbol{x}$ or D' the target translation.

proportion yielding solutions. Langlais and Yvon's [6] so-called T-TRICK exploits the character-count property expressed in (14).

(14) $[A' : B' :: C' : \boldsymbol{x}] \neq \varnothing$ **if** $|A'|_c \leq |B'|_c + |C'|_c \ \ \forall\ c \in \{A',B',C'\}$

H5: Whenever a symbol occurs more frequently in A' than it does in B' and C', the equation is bound to fail and need not be solved.

To this set of heuristics we add our own suggestion, results from which will be reported in the near future. Like H2, we aim to reduce the search space by preferring candidates which are most similar. As a means of solving analogical equations, [14, p.41] notes the property expressed in (15),

(15) $[A : B :: C : D] \Rightarrow \text{dist}(A,B) = \text{dist}(C,D)$ and $\text{dist}(A,C) = \text{dist}(B,D)$

where "dist" is the canonical edit distance measuring insertions and deletions only, not substitutions. This suggests to us a means of speeding up the search process. We first preprocess the entire example database storing the dist score for every pair of examples. Since dist(A,B)=dist(B,A), this involves $n(n-1)/2$ computations for a database with n examples, but can be performed once only, offline.

At runtime, for a given input sentence D we calculate a set of dist(H,D) scores for every H in the database. For each (H,D) pair that we try, we must consider only (A,B) pairs in the matrix that have the same score dist(h,d) as these are the only (A,B) pairs that will contribute to a solvable equation with this H and D. Suppose the first of these is (A_i,B_j), corresponding to dist(i,j) in the matrix. If this represents a solvable equation, we know that dist(i,j)=dist(h,d), and from (15) it therefore follows that dist(i,h)=dist(j,d). We know the value of dist(i,h) because it is precomputed, and we know the value of dist(j,d) because it is among the set of distances that we first calculated. So if dist(i,h)≠dist(j,d) we can immediately reject (A_i,B_j) and try the next pair. In keeping with the spirit of heuristic H2, we can potentially make further savings by sorting the scores for (H,D) pairs, and starting with the lowest.

3.2 Preprocessing

The original "boast" of Lepage and Denoual [10,11,12] is that their approach requires no preprocessing or training of any kind. However, in [13] they abandon this principle, preferring to expand the example data with subsentential (chunk and word) alignments. The chunks are extracted by identifying grammatical markers such as case markers in Japanese. They are then aligned in an example-based process exploiting *hapax legomena* (words occurring exactly once in the corpus) developed by [15]. From original aligned data of 40K Japanese–English and 20K Arabic–English sentence pairs, they were able to extract 84K and nearly 50K chunk alignments respectively, thereby tripling the size of the example set.

Langlais and Yvon [6] introduce a technique by which the input space is organized, presumably offline, which greatly speeds up the search time. It is based on the property, already noted in [8], that there is a symmetry in the count of individual symbols in an analogical equation, as in (16).

(16) $[A : B :: C : D] \Rightarrow |A|_c + |D|_c = |B|_c + |C|_c \ \ \forall\ c \in \{A,B,C,D\}$
where $|S|_c$ is the number of occurrences of the symbol c in the string S.

This means that for any given D, the choice of A puts a constraint on the counts of symbols that any (B, C) pair must satisfy for the analogical equation to hold. This property can lead to a reduced search space if that space is partitioned according to the counts of symbols in each example. Langlais and Yvon [5,6] introduce a tree-count data-structure which does this, and transforms a search of quadratic complexity into a linear one. This approach is only practical if the data is handled as strings of characters.

3.3 Recursion

As mentioned above, it can happen that the straightforward application of the procedure to find a triple (A, B, C) given an input D can lead to a solution involving a string C which is however not itself in the example database. In this circumstance, according to [10,11,12], the translation C' needed to solve the second analogical equation is arrived at "using the present method recursively". This recursion is briefly discussed in [10,12], where it is noted that, in the experiments reported, one recursive call is needed on average, with a maximum of two. There is no explanation or suggestion of

how to control the recursion so as to prevent the system from selecting the same (*A*, *B*) pair as an initial candidate, and thereby getting stuck in a loop. The fact that in their experiments so few recursive calls were needed shows that the sentences in the test set were very close to those in the example set. As Lepage and Denoual state, "the number of recursive calls is a measure of how 'far' a sentence is from a corpus" [12, p.278].

Interestingly, the later GREYC implementations [13,14] omit the recursion stage altogether, with no explanation. In GREYC, when no solution can be found by analogy, the system "backs off to the basic behavior of a translation memory, i.e. it outputs the translation of the source sentence closest to the input sentence."

4 Experiments and results

In this section we review the experiments using this methodology, beginning with the ALEPH system [10,11,12], then the GREYC re-implementation [13,14], followed by a number of experiments where the method is used for the translation of unknown words (only) as part of a broader translation task undertaken using other means [2,4,6,7]. As far as we know, this is the totality of work using this framework for translation to date.

4.1 ALEPH system

The original system, named ALEPH and described in the three 2005 papers [10,11,12], uses as its example base the English, Japanese and Chinese data from the C-STAR project's BTEC, a corpus of around 160K sentences from the travel and tourism domain. Importantly, the average length of the sentences in characters (not words) is only 35.2 (English), 16.2 (Japanese) and 9.4 (Chinese): the examples are "quite short", as the authors freely admit. Some of the sentences appear multiple times, not always with the same translation.

As reported in [10], the system was tested for Japanese–English on a set of 510 sentences from the same domain as the corpus. [11] reports testing on the 2004 IWSLT "unrestricted data" for Japanese–English and Chinese–English, as well as its performance in the IWSLT 2005 competition, including Korean–English (also based on BTEC data), and Arabic–English (using a database of 20K examples). The long paper [12] recaps these results, as well as English–Chinese.

The results are given using a range of evaluation measures, and the system does very well. Compared to the IWSLT 2004 competitors, it comes a close second on all measures for Chinese–English (BLEU score 0.522, for example) and second or third on Japanese–English (BLEU 0.634). In the 2005 campaign, with BLEU scores ranging from 0.382 to 0.593 (discounting a disastrous 0.098 on the English–Chinese task), the system comes 4th or 5th (out of 5 entrants) in the C-STAR-based task [3]. Translation time is on average 0.73 seconds on a 2.8 GHz processor with 4 Gb memory, assuming a time-out of 1 CPU second.

[12] also reports some more details of the performance. With heuristic H1 in place, on average, about 689K equations need to be solved with their corpus of around 100K sentences. The proportion of equations successfully solved is around 28%.

4.2 GREYC implementation

The ALEPH system evolved into a new system, named GREYC, for the 2007 and 2008 IWSLT campaigns [13,14]. The main change was the addition of a preprocessing stage which adds subsentential (word) alignments to the example-base. The double heuristic H2 is added. The system in [14] has an additional refinement of non-determinism. Whereas previous implementations had output only the first solution encountered (a feature by no means made explicit in earlier descriptions), this implementation generates all possible solutions, and is accordingly much slower

Somewhat lower BLEU scores are reported in [13]: 0.396 for Japanese–English based on a 40K example set, and 0.382 for Arabic–English with 20K examples. The authors suggest that the smaller number of reference translations might account for the lower BLEU scores, or at least make comparison with their 2005 system difficult.

The implementation in [14] allows more than one solution but, more significantly, works at the

word rather than character level. It participated in BTEC tasks for Arabic–English, Chinese–English and Chinese–Spanish, as well as the PIVOT task, translating Chinese–Spanish via English, with BLEU scores 0.42, 0.44, 0.23 and 0.21 respectively.

4.3 Using proportional analogies to handle unknown words

While Lepage and colleagues have had modest success using proportional analogies for the full translation task, a similar approach has been reported for the translation of unknown words only, within the context of a statistical MT system means.

Denoual's experiments [2] attempt to translate all unknown words in a Japanese–English task. Training his system on 40K sentence pairs from the IWSLT 2006 competition, supplemented by word-alignment lists extracted using GIZA++. While translation adequacy (as measured by the NIST score) improves, fluency (measured by BLEU) remains stable or is harmed.

Langlais and Patry [4] had rather more success handling unknown words in European languages where parallels in morphological structure can seemingly be exploited, e.g. *activité* : *activity* :: *futilité* : $\boldsymbol{x} \Rightarrow \boldsymbol{x}$ = *futility*. They report that their approach can propose valid translations for around 80% of unknown ordinary words.

Langlais and Yvon [6] use proportional analogies to supplement the word and phrase tables used for standard SMT when a word to be translated is not covered by the statistical model. Experiments involved translating individual words, and phrases of up to five words. The language pair was French–English. All the material (examples and test data) came from the WMT'06 SMT workshop. The example databases themselves were based on phrase tables generated by standard SMT methods, keeping only the five most likely translations. Three sizes of databases were used for each of the two tasks ranging from 57k for the phrase task to 14m for the word task. The authors first investigated various filtering strategies under different conditions, then they report a "front-end evaluation". Even with various filters in place, their methods produce many candidate translations: hundreds, sometimes thousands for phrase translation where the unit of analogy is the word; for the word translation task based on character sequences, the average number of candidate translations was 875,800. The recall rate was 97.5%, but the average position for the first oracle translation was 1,602nd. Clearly some further filtering mechanism on the output is needed. The phrase translation task provided more promising results. Recall is lower (65.2%) and the average number of proposed translations presumably also lower (no figure is given, only the 50 mist frequently proposed are considered), but the average rank for a good translation is 9th. It must be remembered that their goal is to supplement the phrase table in an SMT system, rather than propose a single translation. For this latter task, their approach is clearly unsuitable.

Finally, Langlais et al. [7] applied the method to the translation of medical terms between English and French, Spanish, Finnish and Russian, in both directions. Their results are not so spectacular, but generally showed an improvement on purely statistical approaches.

5 Conclusion and outlook

From a very promising start, as reported in [10,11,12], some of the drawbacks of the proportional analogy approach have since come to light. Unlike other approaches to EBMT, the approach seems to suffer badly when the size of the example base is increased, with both processing times and numbers of solutions increasing. It is clear that heuristics must be introduced to reduce the search space, both in identifying likely (*A*,*B*,*C*) triples, and preventing fruitless attempts to solve equations. And even where equations are solvable, the solutions produced may be in need of filtering. It is apparent that treating the examples as strings of characters leads to excesses in all these areas, while treating them as strings of words leads to an impasse where data is either too sparse to provide a solution, or too big to permit acceptable processing times. A compromise solution in which input is handled as strings of morphemes might be worth trying as a suitable compromise.

While the approach seems fraught with difficulties as a stand-alone translation model, its use for the special case of unknown words, particularly named entities or specialist terms, seems much more promising. We hope to report some results in this area in the near future.

References

[1] Lynne Campbell: *Historical Linguistics: An Overview*, 2nd ed., Cambridge, Massachusetts: The MIT Press, 2004.

[2] Etienne Denoual: Analogical translation of unknown words in a statistical machine translation framework, in *MT Summit XI* (Copenhagen, Denmark, 10-14 September 2007), pp.135-141, 2007.

[3] Matthias Eck and Chiori Hori: Overview of the IWSLT 2005 evaluation campaign, in *International Workshop on Spoken Language Translation: Evaluation Campaign on Spoken Language Translation* [IWSLT 2005] (Pittsburgh, PA, October 24-25, 2005), 22pp, 2005.

[4] Philippe Langlais and Alexandre Patry: Translating unknown words using analogical learning, in *EMNLP-CoNLL-2007: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning* (Prague, Czech Republic, June 28-30, 2007), pp. 877-886, 2007.

[5] Philippe Langlais and François Yvon: Scaling up analogical learning, in *Coling 2008, 22nd International Conference on Computational Linguistics, Companion volume Posters and Demonstrations* (Manchester, UK, 18-22 August, 2008), pp. 51-54, 2008

[6] Philippe Langlais and François Yvon: Scaling up analogical learning, Apprentissage par analogie: passage à l'échelle. Tech. Rep. 2008D014, Telecom ParisTech, Département Informatique et Réseaux, Groupe IC2 : Interaction, Cognition et Complexité, Paris, France, 2008.

[7] Philippe Langlais, François Yvon and Pierre Zweigenbaum: Improvements in analogical learning: application to translating multi-terms of the medical domain, in *EACL-2009: Proceedings of the 12th Conference of the European Chapter of the ACL* (Athens, Greece, 30 March - 3 April, 2009), pp. 487-495, 2009.

[8] Yves Lepage: Solving analogies on words: an algorithm, in *Coling-ACL '98: 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics* (Montreal, Quebec, Canada, August 10-14, 1998), pp. 728-734, 1998.

[9] Yves Lepage: Lower and higher estimates of the number of "true analogies" between sentences contained in a large multilingual corpus, in Coling 2004: 20th International Conference on Computational Linguistics (Geneva, Switzerland, August 23rd-27th, 2004), pp. 736-742, 2004.

[10] Yves Lepage and Etienne Denoual: The 'purest' EBMT system ever built: no variables, no templates, no training, examples, just examples, only examples, in *MT Summit X, Second Workshop on Example-Based Machine Translation* (Phuket, Thailand, Sept. 12-16, 2005), pp. 81-90, 2005.

[11] Yves Lepage and Etienne Denoual: ALEPH: an EBMT system based on the preservation of proportional analogies between sentences across languages in *International Workshop on Spoken Language Translation: Evaluation Campaign on Spoken Language Translation* [IWSLT 2005] (Pittsburgh, PA, Oct. 24-25, 2005), 8pp, 2005.

[12] Yves Lepage and Etienne Denoual: Purest ever example-based machine translation: Detailed presentation and assessment, *Machine Translation* 19:251-282, 2005.

[13] Yves Lepage and Adrien Lardilleux: The GREYC machine translation system for the IWSLT 2007 evaluation campaign, in *IWSLT 2007: International Workshop on Spoken Language Translation* (Trento, Italy, 15-16 October 2007), 7pp, 2007.

[14] Yves Lepage, Adrien Lardilleux, Julien Gosme and Jean-Luc Manguin: The GREYC machine translation system for the IWSLT 2008 evaluation campaign, in *IWSLT 2008: Proceedings of the International Workshop on Spoken Language Translation* (Hawaii, USA, 20-21 October 2008), pp. 39-45, 2008.

[15] Yves Lepage, Julien Migeot and Erwan Guillerm: Analogies of form between chunks in Japanese are massive and far from being misleading, in *LTC'07: 3rd Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics* (Poznań, Poland, October 5-7, 2007), pp. 503-507, 2007.