Improvements in Analogical Learning: Application to Translating multi-Terms of the Medical Domain

Philippe Langlais
DIRO
Univ. of Montreal, Canada
felipe@iro.umontreal.ca

François Yvon and Pierre Zweigenbaum
LIMSI-CNRS
Univ. Paris-Sud XI, France
{yvon,pz}@limsi.fr

Abstract

Handling terminology is an important matter in a translation workflow. However, current Machine Translation (MT) systems do not yet propose anything proactive upon tools which assist in managing terminological databases. In this work, we investigate several enhancements to analogical learning and test our implementation on translating medical terms. We show that the analogical engine works equally well when translating from and into a morphologically rich language, or when dealing with language pairs written in different scripts. Combining it with a phrase-based statistical engine leads to significant improvements.

1 Introduction

If machine translation is to meet commercial needs, it must offer a sensible approach to translating terms. Currently, MT systems offer at best database management tools which allow a human (typically a translator, a terminologist or even the vendor of the system) to specify bilingual terminological entries. More advanced tools are meant to identify inconsistencies in terminological translations and might prove useful in controlled-language situations (Itagaki et al., 2007).

One approach to translate terms consists in using a domain-specific parallel corpus with standard alignment techniques (Brown et al., 1993) to *mine* new translations. Massive amounts of parallel data are certainly available in several pairs of languages for domains such as parliament debates or the like. However, having at our disposal a domain-specific (*e.g.* computer science) bitext with an adequate coverage is another issue. One might argue that domain-specific comparable (or perhaps unrelated) corpora are easier to acquire, in which case context-vector techniques (Rapp, 1995; Fung and McKeown, 1997) can be used to *identify* the translation of terms. We certainly agree with that point of view to a certain extent, but as discussed by Morin et al. (2007), for many specific domains and pairs of languages, such resources simply do not exist. Furthermore, the task of translation identification is more difficult and error-prone.

Analogical learning has recently regained some interest in the NLP community. Lepage and Denoual (2005) proposed a machine translation system entirely based on the concept of *formal analogy*, that is, analogy on forms. Stroppa and Yvon (2005) applied analogical learning to several morphological tasks also involving analogies on words. Langlais and Patry (2007) applied it to the task of translating unknown words in several European languages, an idea investigated as well by Denoual (2007) for a Japanese to English translation task.

In this study, we improve the state-of-the-art of analogical learning by (i) proposing a simple yet effective implementation of an analogical solver; (ii) proposing an efficient solution to the search issue embedded in analogical learning, (iii) investigating whether a classifier can be trained to recognize bad candidates produced by analogical learning. We evaluate our analogical engine on the task of translating terms of the medical domain; a domain well-known for its tendency to create new words, many of which being complex lexical constructions. Our experiments involve five language pairs, including languages with very different morphological systems.

Proceedings of the 12th Conference of the European Chapter of the ACL, pages 487–495,
Athens, Greece, 30 March – 3 April 2009.

In the remainder of this paper, we first present in Section 2 the principle of analogical learning. Practical issues in analogical learning are discussed in Section 3 along with our solutions. In Section 4, we report on experiments we conducted with our analogical device. We conclude this study and discuss future work in Section 5.

2 Analogical Learning

2.1 Definitions

A *proportional analogy*, or analogy for short, is a relation between four items noted $[x : y = z : t]$ which reads as "*x* is to *y* as *z* is to *t*". Among proportional analogies, we distinguish *formal analogies*, that is, those we can identify at a graphemic level, such as [*adrenergic beta-agonists*, *adrenergic beta-antagonists*, *adrenergic alpha-agonists*, *adrenergic alpha-antagonists*].

Formal analogies can be defined in terms of factorizations[1]. Let *x* be a string over an alphabet Σ, a *factorization* of *x*, noted f_X, is a sequence of n *factors* $f_X = (f_X^1, \ldots, f_X^n)$, such that $x = f_X^1 \odot f_X^2 \odot \ldots \odot f_X^n$, where $\odot$ denotes the concatenation operator. After (Stroppa and Yvon, 2005) we thus define a formal analogy as:

Definition 1 $\forall (x, y, z, t) \in \Sigma^{\star 4}$, $[x : y = z : t]$ ***iff*** *there exist factorizations* $(f_X, f_Y, f_Z, f_t) \in (\Sigma^{\star^d})^4$ *of* (x, y, z, t) *such that,* $\forall i \in [1, d]$, $(f_Y^i, f_Z^i) \in \left\{(f_X^i, f_t^i), (f_t^i, f_X^i)\right\}$. *The smallest* d *for which this definition holds is called the degree of the analogy.*

Intuitively, this definition states that (x, y, z, t) are made up of a common set of alternating substrings. It is routine to check that it captures the exemplar analogy introduced above, based on the following set of factorizations:

$$\begin{aligned} f_X &\equiv (\text{adrenergic bet}, \text{a-agonists}) \\ f_Y &\equiv (\text{adrenergic bet}, \text{a-antagonists}) \\ f_Z &\equiv (\text{adrenergic alph}, \text{a-agonists}) \\ f_t &\equiv (\text{adrenergic alph}, \text{a-antagonists}) \end{aligned}$$

As no smaller factorization can be found, the degree of this analogy is 2. In the sequel, we call an *analogical equation* an analogy where one item (usually the fourth) is missing and we note it $[x : y = z : ?]$.

[1]Factorizations of strings correspond to segmentations. We keep the former term, to emphasize the genericity of the definition, which remains valid for other algebraic structures, for which factorization and segmentation are no longer synonymous.

2.2 Analogical Inference

Let $\mathcal{L} = \{(i, o) \,|\, i \in \mathcal{I}, o \in \mathcal{O}\}$ be a learning set of observations, where $\mathcal{I}$ ($\mathcal{O}$) is the set of possible forms of the input (output) linguistic system under study. We denote $I(u)$ ($O(u)$) the projection of u into the input (output) space; that is, if $u = (i, o)$, then $I(u) \equiv i$ and $O(u) \equiv o$. For an incomplete observation $u = (i, ?)$, the inference procedure is:

1. building $\mathcal{E}_\mathcal{I}(u) = \{(x, y, z) \in \mathcal{L}^3 \mid [I(x) : I(y) = I(z) : I(u)]\}$, the set of input triplets that define an analogy with $I(u)$.
2. building $\mathcal{E}_\mathcal{O}(u) = \{o \in \mathcal{O} \mid \exists (x, y, z) \in \mathcal{E}_\mathcal{I}(u)$ s.t. $[O(x) : O(y) = O(z) : o]\}$ the set of solutions to the equations obtained by projecting the triplets of $\mathcal{E}_\mathcal{I}(u)$ into the output space.
3. selecting candidates among $\mathcal{E}_\mathcal{O}(u)$.

In the sequel, we distinguish the *generator* which implements the first two steps, from the *selector* which implements step 3.

To give an example, assume $\mathcal{L}$ contains the following entries: *(beeta-agonistit, adrenergic beta-agonists)*, *(beetasalpaajat, adrenergic beta-antagonists)* and *(alfa-agonistit, adrenergic alpha-agonists)*. We might translate the Finnish term *alfasalpaajat* into the English term *adrenergic alpha-antagonists* by 1) identifying the input triplet: *(beeta-agonistit, beetasalpaajat, alfa-agonistit)*; 2) projecting it into the equation [*adrenergic beta-agonists* : *adrenergic beta-antagonists* = *adrenergic alpha-agonists* : ?]; and solving it: *adrenergic alpha-antagonists* is one of its solutions.

During inference, analogies are recognized independently in the input and the output space, and nothing pre-establishes which subpart of one input form corresponds to which subpart of the output one. This "knowledge" is passively captured thanks to the inductive bias of the learning strategy (an analogy in the input space corresponds to one in the output space). Also worth mentioning, this procedure does not rely on any pre-defined notion of word. This might come at an advantage for languages that are hard to segment (Lepage and Lardilleux, 2007).

3 Practical issues

Each step of analogical learning, that is, *searching* for input triplets, *solving* output equations and

selecting good candidates involves some practical issues. Since searching for input triplets might involve the need for solving (input) equations, we discuss the solver first.

3.1 The solver

Lepage (1998) proposed an algorithm for solving an analogical equation $[x : y = z : ?]$. An alignment between *x* and *y* and between *x* and *z* is first computed (by edit-distance) as illustrated in Figure 1. Then, the three strings are synchronized using *x* as a backbone of the synchronization. The algorithm can be seen as a deterministic finite-state machine where a state is defined by the two edit-operations being visited in the two tables. This is schematized by the two cursors in the figure. Two actions are allowed: `copy` one symbol from *y* or *z* into the solution and `move` one or both cursors.

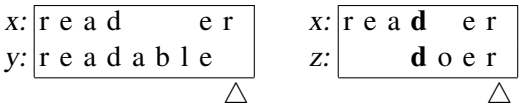

Figure 1: Illustration of the synchronization done by the solver described in (Lepage, 1998).

There are two things to realize with this algorithm. First, since several (minimal-cost) alignments can be found between two strings, several synchronizations are typically carried out while solving an equation, leading to (possibly many) different solutions. Indeed, in adverse situations, an exponential number of synchronizations will have to be computed. Second, the algorithm fails to deliver an expected form in a rather frequent situation where two identical symbols align fortuitously in two strings. This is for instance the case in our running example where the symbol *d* in *doer* aligns to the one in *reader*, which puzzles the synchronization. Indeed, *dabloe* is the only form proposed to $[$*reader* : *readable* = *doer* : $?]$, while the expected one is *doable*. The algorithm would have no problem, however, to produce the form *writable* out of the equation $[$*reader* : *readable* = *writer* : $?]$.

Yvon et al. (2004) proposed an analogical solver which is not exposed to the latter problem. It consists in building a finite state transducer which generates the solutions to $[x : y = z : ?]$ while recognizing the form *x*.

Theorem 1 *t is a solution to* $[x : y = z : ?]$ ***iff*** *t belongs to* $\{y \circ z\} \backslash x$.

shuffle and *complement* are two rational operations. The *shuffle* of two strings *w* and *v*, noted $w \circ v$, is the regular language containing the strings obtained by selecting (without replacement) alternatively in *w* and *v*, sequences of characters in a left-to-right manner. For instance, $\underline{spondy}ondonti\underline{lalgia}tis$ and $ond\underline{spond}onyla\underline{ltitisgia}$ are two strings belonging to $\underline{spondylalgia} \circ ondontitis$). The *complementary set* of *w* with respect to *v*, noted $w \backslash v$, is the set of strings formed by removing from *w*, in a left-to-right manner, the symbols in *v*. For instance, $spondylitis$ and $spydoniltis$ are belonging to $spondyondontilalgiatis \setminus ondontalgia$. Our implementation of the two rational operations are sketched in Algorithm 1.

Because the shuffle of two strings may contain an exponential number of elements with respect to the length of those strings, building such an automaton may face combinatorial problems. Our solution simply consists in randomly sampling strings in the shuffle set. Our solver, depicted in Algorithm 2, is thus controlled by a sampling size `s`, the impact of which is illustrated in Table 1. By increasing `s`, the solver generates more (mostly spurious) solutions, but also increases the relative frequency with which the expected output is generated. In practice, provided a large enough sampling size,[2] the expected form very often appears among the most frequent ones.

s	nb	(solution,frequency)		
10	11	(doable,7)	(dabloe,3)	(adbloe,3)
10^2	22	(doable,28)	(dabloe,21)	(abldoe,21)
10^3	29	(doable,333)	(dabloe,196)	(abldoe,164)

Table 1: The 3-most frequent solutions generated by our solver, for different sampling sizes `s`, for the equation $[$*reader* : *readable* = *doer* : $?]$. nb indicates the number of (different) solutions generated. According to our definition, there are 32 distinct solutions to this equation. Note that our solver has no problem producing *doable*.

3.2 Searching for input triplets

A brute-force approach to identifying the input triplets that define an analogy with the incomplete observation $u = (t, ?)$ consists in enumerating triplets in the input space and checking for an

[2] We used `s` = 2 000 in this study.

```
function shuffle(y,z)
  Input: ⟨y, z⟩ two forms
  Output: a random word in y ∘ z
  if y = ϵ then
    return z
  else
    n ← rand(1,|y|)
    return y[1:n] . shuffle(z,y[n+1:])

function complementary(m,x,r,s)
  Input: m ∈ y ∘ z, x
  Output: the set m \ x
  if (m = ϵ) then
    if (x = ϵ) then
      s ← s ∪ r
  else
    complementary(m[2:],x,r.m[1],s)
    if m[1] = x[1] then
      complementary(m[2:],x[2:],r,s)
```

Algorithm 1: Simulation of the two rational operations required by the solver. $x[a{:}b]$ denotes the sequence of symbols x starting from index a to index b inclusive. $x[a{:}]$ denotes the suffix of x starting at index a.

analogical relation with t. This amounts to check $o(|\mathcal{I}|^3)$ analogies, which is manageable for toy problems only. Instead, Langlais and Patry (2007) proposed to solve analogical equations $[y : x = t : ?]$ for some pairs $\langle x, y\rangle$ belonging to the neighborhood[3] of $I(u)$, denoted $\mathcal{N}(t)$. Those solutions that belong to the input space are the z-forms retained;

$$\mathcal{E}_{\mathcal{I}}(u) = \{\ \langle x, y, z\rangle\ :\ \ x \in \mathcal{N}(t)\ ,\ y \in \mathcal{N}(x),\ z \in [y : x = t : ?] \cap \mathcal{I}\ \}$$

This strategy (hereafter named LP) directly follows from a symmetrical property of an analogy ($[x : y = z : t] \Leftrightarrow [y : x = t : z]$), and reduces the search procedure to the resolution of a number of analogical equations which is quadratic with the number of pairs $\langle x, y\rangle$ sampled.

We found this strategy to be of little use for input spaces larger than a few tens of thousands forms. To solve this problem, we exploit a property on symbol counts that an analogical relation must fulfill (Lepage, 1998):

$$[x : y = z : t] \Rightarrow |x|_c + |t|_c = |y|_c + |z|_c \quad \forall c \in \mathcal{A}$$

[3]The authors proposed to sample x and y among the closest forms in terms of edit-distance to $I(u)$.

```
function solver(⟨x, y, z⟩, s)
  Input: ⟨x, y, z⟩, a triplet, s the sampling size
  Output: a set of solutions to [x : y = z : ?]
  sol ← ϕ
  for i ← 1 to s do
    ⟨a, b⟩ ← odd(rand(0, 1))? ⟨z, y⟩ : ⟨y, z⟩
    m ← shuffle(a,b)
    c ← complementary(m,x,ϵ,{})
    sol ← sol ∪ c
  return sol
```

Algorithm 2: A Stroppa&Yvon flavored solver. $\texttt{rand}(a, b)$ returns a random integer between a and b (included). The ternary operator `?:` is to be understood as in the C language.

where $\mathcal{A}$ is the alphabet on which the forms are built, and $|x|_c$ stands for the number of occurrences of symbol c in x.

Our search strategy (named TC) begins by selecting an x-form in the input space. This enforces a set of necessary constraints on the counts of characters that any two forms y and z must satisfy for $[x : y = z : t]$ to be true. By considering all forms x in turn,[4] we collect a set of candidate triplets for t. A verification of those that define with t an analogy must then be carried out. Formally, we built:

$$\mathcal{E}_{\mathcal{I}}(u) = \{\ \langle x, y, z\rangle\ :\ \ x \in \mathcal{I},\ \langle y, z\rangle \in \mathcal{C}(\langle x, t\rangle),\ [x : y = z : t]\ \}$$

where $\mathcal{C}(\langle x, t\rangle)$ denotes the set of pairs $\langle y, z\rangle$ which satisfy the count property.

This strategy will only work if (i) the number of quadruplets to check is much smaller than the number of triplets we can form in the input space (which happens to be the case in practice), and if (ii) we can efficiently identify the pairs $\langle y, z\rangle$ that satisfy a set of constraints on character counts. To this end, we proposed in (Langlais and Yvon, 2008) to organize the input space into a data structure which supports efficient runtime retrieval.

3.3 The selector

Step 3 of analogical learning consists in selecting one or several solutions from the set of candidate forms produced by the generator. We trained in a supervised manner a binary classifier to distinguish good translation candidates (as defined by

[4]Anagram forms do not have to be considered separately.

a reference) from spurious ones. We applied to this end the *voted-perceptron* algorithm described by Freund and Schapire (1999). Online voted-perceptrons have been reported to work well in a number of NLP tasks (Collins, 2002; Liang et al., 2006). Training such a classifier is mainly a matter of feature engineering. An *example* e is a pair of source-target analogical relations $(r, \hat{r})$ identified by the generator, and which elects $\hat{t}$ as a translation for the term *t*:

$$e \equiv (r, \hat{r}) \equiv ([x : y = z : t], [\hat{x} : \hat{y} = \hat{z} : \hat{t}])$$

where $\hat{x}$, $\hat{y}$, and $\hat{z}$ are respectively the projections of the source terms *x*, *y* and *z*. We investigated many features including (i) the degree of r and $\hat{r}$, (ii) the frequency with which a form is generated,[5] (iii) length ratios between t and $\hat{t}$, (iv) likelihoods scores (min, max, avg.) computed by a character-based n-gram model trained on a large general corpus (without overlap to DEV or TRAIN), etc.

4 Experiments

4.1 Calibrating the engine

We compared the two aforementioned searching strategies on a task of identifying triplets in an input space of French words for 1 000 randomly selected test words. We considered input spaces of various sizes. The results are reported in Table 2. TC clearly outperforms LP by systematically identifying more triplets in much less time. For the largest input space of 84 000 forms, TC could identify an average of 746 triplets for 946 test words in 1.2 seconds, while the best compromise we could settle with LP allows the identification of 56 triplets on average for 889 words in 6.3 seconds on average. Note that in this experiment, LP was calibrated for each input space so that the best compromise between recall ($\%s$) and speed could be found. Reducing the size of the neighborhood in LP improves computation time, but significantly affects recall. In the following, we only consider the TC search strategy.

4.2 Experimental Protocol

Datasets The data we used in this study comes from the Medical Subject Headings (MeSH) thesaurus. This thesaurus is used by the US National Library of Medicine to index the biomedical scientific literature in the MEDLINE database.[6] Its preferred terms are called "Main Headings". We collected pairs of source and target Main Headings (TTY = 'MH') with the same MeSH identifiers (SDUI).

We considered five language pairs with three relatively close European languages (English-French, English-Spanish and English-Swedish), a more distant one (English-Finnish) and one pair involving different scripts (English-Russian).[7]

The material was split in three randomly selected parts, so that the development and test material contain exactly 1 000 terms each. The characteristics of this material are reported in Table 3. For the Finnish-English and Swedish-English language pairs, the ratio of uni-terms in the Foreign language ($u_f\%$) is twice the ratio of uni-terms in the English counterpart. This is simply due to the agglutinative nature of these two languages. For instance, according to MeSH, the English multi-term *speech articulation tests* corresponds to the Finnish uni-term *ääntämiskokeet* and to the Swedish one *artikulationstester*. The ratio of out-of-vocabulary forms (space-separated words unseen in TRAIN) in the TEST material is rather high: between 36% and 68% for all Foreign-to-English translation directions, but Finnish-to-English, where surprisingly, only 6% of the word forms are unknown.

	s	$\%s$	(s)	s	$\%s$	(s)	s	$\%s$	(s)
TC	34	83.1	0.2	261	94.1	0.5	746	96.4	1.2
LP	17	71.7	7.4	46	85.0	7.6	56	88.9	6.3
$\|\mathcal{I}\|$	20 000			50 000			84 076		

Table 2: Average number s of input analogies found over 1 000 test words as a function of the size of the input space. $\%s$ stands for the percentage of source forms for which (at least) one source triplet is found; and (s) indicates the average time (counted in seconds) to treat one form.

Evaluation metrics For each experimental condition, we compute the following measures:

Coverage the fraction of input words for which the system can generate translations. If N_t words receive translations among N, coverage is N_t/N.

[5] A form $\hat{t}$ may be generated thanks to many examples.

[6] The MeSH thesaurus and its translations are included in the UMLS Metathesaurus.

[7] Russian MeSH is normally written in Cyrillic, but some terms are simply English terms written in uppercase Latin script (e.g., *ACHROMOBACTER* for English *Achromobacter*). We removed those terms.

	TRAIN			TEST	DEV	TEST
f	nb	u_f%	u_e%	nb	u_f%	u_f%	oov%
FI	19 787	63.7	33.7	1 000	64.2	64.0	5.7
FR	17 230	29.8	29.3	1 000	30.8	28.3	36.3
RU	21 407	38.6	38.6	1 000	38.5	40.2	44.4
SP	19 021	31.1	31.1	1 000	31.7	33.3	36.6
SW	17 090	67.9	32.5	1 000	67.4	67.9	68.4

Table 3: Main characteristics of our datasets. nb indicates the number of pairs of terms in a bitext, u_f% (u_e%) stands for the percentage of uniterms in the *Foreign* (English) part. oov% indicates the percentage of out-of-vocabulary forms (space-separated forms of TEST unseen in TRAIN).

	Cov	P_1	R_1	P_{100}	R_{100}	R_∞
→ FI	**47.1**	*31.6*	14.9	*57.7*	27.2	31.9
FR	41.2	35.4	*14.6*	60.4	24.9	*26.5*
RU	46.2	40.5	18.7	69.9	32.3	34.8
SP	47.0	41.5	19.5	69.1	**32.5**	**35.9**
SW	42.8	36.0	15.4	66.8	28.6	31.9
← FI	44.8	36.6	16.4	66.7	29.9	33.2
FR	*38.5*	47.0	18.1	69.9	*26.9*	29.4
RU	42.1	**49.4**	**20.8**	70.3	29.6	32.3
SP	42.6	47.7	20.3	**75.1**	32.0	33.7
SW	44.6	40.8	18.2	69.5	31.0	32.9

Table 4: Main characteristics of the generator, as a function of the translation directions (TEST).

Precision among the N_t words for which the system proposes an answer, precision is the proportion of those for which a correct translation is output. Depending on the number of output translations k that one is willing to examine, a correct translation will be output for N_k input words. Precision at rank k is thus defined as $P_k = N_k/N_t$.

Recall is the proportion of the N input words for which a correct translation is output. Recall at rank k is defined as $R_k = N_k/N$.

In all our experiments, candidate translations are sorted in decreasing order of frequency with which they were generated.

4.3 The generator

The performances of the generator on the 10 translation sessions are reported in Table 4. The coverage of the generator varies between 38.5% (French-to-English) and 47.1% (English-to-Finnish), which is rather low. In most cases, the silence of the generator is due to a failure to identify analogies in the input space (step 1). The last column of Table 4 reports the maximum recall we can obtain if we consider all the candidates output by the generator. The relative accuracy of the generator, expressed by the ratio of R_∞ to cov, ranges from 64.3% (English-French) to 79.1% (Spanish-to-English), for an average value of 73.8% over all translation directions. This roughly means that one fourth of the test terms with at least one solution do not contain the reference.

Overall, we conclude that analogical learning offers comparable performances for all translation directions, although some fluctuations are observed. We do not observe that the approach is affected by language pairs which do not share the same script (Russian/English). The best (worse) case (as far as R_∞ is concerned) corresponds to translating into Spanish (French).

Admittedly, the largest recall and R_∞ values reported in Table 4 are disappointing. Clearly, for analogical learning to work efficiently, enough linguistic phenomena must be attested in the TRAIN material. To illustrate this, we collected for the Spanish-English language pair a set of medical terms from the Medical Drug Regulatory Activities thesaurus (MedDRA) which contains roughly three times more terms than the Spanish-English material used in this study. This extra material allows to raise the coverage to 73.4% (Spanish to English) and 79.7% (English to Spanish), an absolute improvement of more than 30%.

4.4 The selector

We trained our classifiers on the several millions of *examples* generated while translating the development material. Since we considered numerous feature representations in this study, this implies saving many huge datafiles on disk. In order to save some space, we decided to remove forms that were generated less than 3 times.[8] Each classifier was trained using 20 epochs.

It is important to note that we face a very unbalanced task. For instance, for the English to Finnish task, the generator produces no less than 2.7 millions of examples, among which only 4 150 are positive ones. Clearly, classifying all the examples as negative will achieve a very high classification accuracy, but will be of no practical use. Therefore, we measure the ability of a classifier to iden-

[8] Averaged over all translation directions, this incurs an absolute reduction of the coverage of 3.4%.

	FI→EN		FR→EN		RU→EN		SP→EN		SW→EN	
	p	r	p	r	p	r	p	r	p	r
`argmax-f1`	41.3	56.7	46.7	63.9	48.1	65.6	49.2	63.4	43.2	61.0
`s-best`	53.6	61.3	57.5	68.4	61.9	66.7	64.3	70.0	53.1	64.4

Table 5: Precision (p) and recall (r) of some classifiers on the TEST material.

tify the few positive *forms* among the set of candidates. We measure *precision* as the percentage of forms selected by the classifier that are sanctioned by the reference lexicon, and *recall* as the percentage of forms selected by the classifier over the total number of sanctioned forms that the classifier could possibly select. (Recall that the generator often fails to produce oracle forms.)

The performance measured on the TEST material of the best classifier we monitored on DEV are reported in Table 5 for the Foreign-to-English translation directions (we made consistent observations on the reverse directions). For comparison purposes, we implemented a baseline classifier (lines `argmax-f1`) which selects the most-frequent candidate form. This is the selector used as a default in several studies on analogical learning (Lepage and Denoual, 2005; Stroppa and Yvon, 2005). The baseline identifies between 56.7% to 65.6% of the sanctioned forms, at precision rates ranging from 41.3% to 49.2%. We observe for all translation directions that the best classifier we trained systematically outperforms this baseline, both in terms of precision and recall.

4.4.1 The overall system

Table 6 shows the overall performance of the analogical translation device in terms of precision, recall and coverage rates as defined in Section 4.2. Overall, our best configuration (the one embedding the `s-best` classifier) translates between 19.3% and 22.5% of the test material, with a precision ranging from 50.4% to 63.2%. This is better than the variant which always proposes the most frequent generated form (`argmax-f1`). Allowing more answers increases both precision and recall. If we allow up to 10 candidates per source term, the analogical translator translates one fourth of the terms (26.1%) with a precision of 70.9%, averaged over all translation directions. The `oracle` variant, which looks at the reference for selecting the good candidates produced by the generator, gives an upper bound of the performance that could be obtained with our approach: less than a third of the source terms can be translated correctly. Recall however that increasing the TRAIN material leads to drastic improvements in coverage.

4.5 Comparison with a PB-SMT engine

To put these figures in perspective, we measured the performance of a phrase-based statistical MT (PB-SMT) engine trained to handle the same translation task. We trained a phrase table on TRAIN, using the standard approach.[9] However, because of the small training size, and the rather huge OOV rate of the translation tasks we address, we did not train translation models on word-tokens, but at the character level. Therefore a phrase is indeed a sequence of characters. This idea has been successively investigated in a Catalan-to-Spanish translation task by Vilar et al. (2007). We tuned the 8 coefficients of the so-called log-linear combination maximized at decoding time on the first 200 pairs of terms of the DEV corpora. On the DEV set, BLEU scores[10] range from 67.2 (English-to-Finnish) to 77.0 (Russian-to-English).

Table 7 reports the precision and recall of both translation engines. Note that because the SMT engine always propose a translation, its precision equals its recall. First, we observe that the precision of the SMT engine is not high (between 17% and 31%), which demonstrates the difficulty of the task. The analogical device does better for all translation directions (see Table 6), but at a much lower recall, remaining silent more than half of the time. This suggests that combining both systems could be advantageous. To verify this, we ran a straightforward combination: whenever the analogical device produces a translation, we pick it; otherwise, the statistical output is considered. The gains of the resulting system over the SMT alone are reported in column ΔB. Averaged over

[9] We used the scripts distributed by Philipp Koehn to train the phrase-table, and `Pharaoh` (Koehn, 2004) for producing the translations.

[10] We computed BLEU scores at the character level.

		FI→EN		FR→EN		RU→EN		SP→EN		SW→EN	
	k	P_k	R_k	P_k	R_k	P_k	R_k	P_k	R_k	P_k	R_k
`argmax-f`	1	41.3	17.3	46.7	16.8	47.8	18.6	48.7	19.2	43.4	18.1
	10	61.6	25.8	62.8	22.6	61.7	24.0	69.3	27.3	62.1	25.9
`s-best`	1	53.5	20.8	56.9	19.3	58.5	20.3	63.2	22.5	50.4	21
	10	69.4	27.0	69.0	23.4	71.8	24.9	78.4	27.9	65.7	27.4
oracle	1	100	30.5	100	26.3	100	28.5	100	30.6	100	29.5

Table 6: Precision and recall at rank 1 and 10 for the Foreign-to-English translation tasks (TEST).

all translation directions, BLEU scores increase on TEST from 66.2 to 71.5, that is, an absolute improvement of 5.3 points.

	→ EN		← EN	
	P_{smt}	ΔB	P_{smt}	ΔB
FI	20.2	+7.4	21.6	+6.4
FR	19.9	+5.3	17.0	+6.0
RU	24.1	+3.1	28.0	+6.4
SP	22.1	+4.9	26.4	+5.5
SW	25.9	+4.2	31.6	+3.2

Table 7: Translation performances on TEST. P_{smt} stands for the precision and recall of the SMT engine. ΔB indicates the absolute gain in BLEU score of the combined system.

We noticed a tendency of the statistical engine to produce literal translations; a default the analogical device does not show. For instance, the Spanish term *instituciones de atención ambulatoria* is translated word for word by `Pharaoh` into *institutions, atention ambulatory* while analogical learning produces *ambulatory care facilities*. We also noticed that analogical learning sometimes produces wrong translations based on morphological regularities that are applied blindly. This is, for instance, the case in a Russian/English example where *mouthal manifestations* is produced, instead of *oral manifestations*.

5 Discussion and future work

In this study, we proposed solutions to practical issues involved in analogical learning. A simple yet effective implementation of a solver is described. A search strategy is proposed which outperforms the one described in (Langlais and Patry, 2007). Also, we showed that a classifier trained to select good candidate translations outperforms the *most-frequently-generated* heuristic used in several works on analogical learning.

Our analogical device was used to translate medical terms in different language pairs. The approach rates comparably across the 10 translation directions we considered. In particular, we do not see a drop in performance when translating into a morphology rich language (such as Finnish), or when translating into languages with different scripts. Averaged over all translation directions, the best variant could translate in first position 21% of the terms with a precision of 57%, while at best, one could translate 30% of the terms with a perfect precision. We show that the analogical translations are of better quality than those produced by a phrase-based engine trained at the character level, albeit with much lower recall. A straightforward combination of both approaches led an improvement of 5.3 BLEU points over the SMT alone. Better SMT performance could be obtained with a system based on morphemes, see for instance (Toutanova et al., 2008). However, since lists of morphemes specific to the medical domain do not exist for all the languages pairs we considered here, unsupervised methods for acquiring morphemes would be necessary, which is left as a future work. In any case, this comparison is meaningful, since both the SMT and the analogical device work at the character level.

This work opens up several avenues. First, we will test our approach on terminologies from different domains, varying the size of the training material. Second, analyzing the segmentation induced by analogical learning would be interesting. Third, we need to address the problem of combining the translations produced by analogy into a front-end statistical translation engine. Last, there is no reason to constrain ourselves to translating terminology only. We targeted this task in the first place, because terminology typically plugs translation systems, but we think that analogical learning could be useful for translating infrequent entities.

References

P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

M. Collins. 2002. Discriminative training methods for hidden markov models: theory and experiments with perceptron algorithms. In *EMNLP*, pages 1–8, Morristown, NJ, USA.

E. Denoual. 2007. Analogical translation of unknown words in a statistical machine translation framework. In *MT Summit, XI*, pages 10–14, Copenhagen.

Y. Freund and R. E. Schapire. 1999. Large margin classification using the perceptron algorithm. *Mach. Learn.*, 37(3):277–296.

P. Fung and K. McKeown. 1997. Finding terminology translations from non-parallel corpora. In *5th Annual Workshop on Very Large Corpora*, pages 192–202, Hong Kong.

M. Itagaki, T. Aikawa, and X. He. 2007. Automatic validation of terminology translation consistency with statistical method. In *MT Summit XI*, pages 269–274, Copenhagen, Denmark.

P. Koehn. 2004. Pharaoh: A beam search decoder for phrase-based statistical machine translation models. In *AMTA*, pages 115–124, Washington, DC, USA.

P. Langlais and A. Patry. 2007. Translating unknown words by analogical learning. In *EMNLP-CoNLL*, pages 877–886, Prague, Czech Republic.

P. Langlais and F. Yvon. 2008. Scaling up analogical learning. In *22nd International Conference on Computational Linguistics (COLING 2008)*, pages 51–54, Manchester, United Kingdom.

Y. Lepage and E. Denoual. 2005. ALEPH: an EBMT system based on the preservation of proportionnal analogies between sentences across languages. In *International Workshop on Statistical Language Translation (IWSLT)*, Pittsburgh, PA, October.

Y. Lepage and A. Lardilleux. 2007. The GREYC Machine Translation System for the IWSLT 2007 Evaluation Campaign. In *IWLST*, pages 49–53, Trento, Italy.

Y. Lepage. 1998. Solving analogies on words: an algorithm. In *COLING-ACL*, pages 728–734, Montreal, Canada.

P. Liang, A. Bouchard-Côté, D. Klein, and B. Taskar. 2006. An end-to-end discriminative approach to machine translation. In *21st COLING and 44th ACL*, pages 761–768, Sydney, Australia.

E. Morin, B. Daille, K. Takeuchi, and K. Kageura. 2007. Bilingual terminology mining - using brain, not brawn comparable corpora. In *45th ACL*, pages 664–671, Prague, Czech Republic.

R. Rapp. 1995. Identifying word translation in non-parallel texts. In *33rd ACL*, pages 320–322, Cambridge,Massachusetts, USA.

N. Stroppa and F. Yvon. 2005. An analogical learner for morphological analysis. In *9th CoNLL*, pages 120–127, Ann Arbor, MI.

K Toutanova, H. Suzuki, and A. Ruopp. 2008. Applying morphology generation models to machine translation. In *ACL-8 HLT*, pages 514–522, Colombus, Ohio, USA.

D. Vilar, J. Peter, and H. Ney. 2007. Can we translate letters? In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 33–39, Prague, Czech Republic, June.

F. Yvon, N. Stroppa, A. Delhay, and L. Miclet. 2004. Solving analogical equations on words. Technical Report D005, École Nationale Supérieure des Télécommunications, Paris, France, July.