Fourteen Light Tasks for Comparing Analogical and Phrase-based Machine Translation

Rafik Rhouma
RALI / DIRO
Université de Montréal
rafikrhouma@live.fr

Philippe Langlais
RALI / DIRO
Université de Montréal
felipe@iro.umontreal.ca

Abstract

In this study we compare two machine translation devices on twelve machine translation medical-domain specific tasks, and two transliteration tasks, altogether involving twelve language pairs, including English-Chinese and English-Russian, which do not share the same scripts. We implemented an analogical device and compared its performance to the state-of-the-art phrase-based machine translation engine Moses. On most translation tasks, the analogical device outperforms the phrase-based one, and several combinations of both systems significantly outperform each system individually. For the sake of reproducibility, we share the datasets used in this study.

1 Introduction

A *proportional analogy* is a relation between 4 objets, x, y, z and t, noted $[x : y :: z : t]$, which reads x is to y as z is to t. A *formal proportional analogy*, hereafter analogy, is a proportional analogy which involves a relationship at the graphemic level, such as [*atomkraftwerken* : *atomkriegen* :: *kraftwerks* : *kriegs*] in German. *Analogical learning* is a holistic learning paradigm (sketched in Section 2) which relies on proportional analogies for generalizing a training set.

Lepage and Denoual (2005b) pioneered the application of analogical learning to Machine Translation (MT). Different variants of their system have been tested within the IWSLT evaluation campaigns (Lepage and Denoual, 2005a; Lepage and Lardilleux, 2008; Lepage et al., 2008; Lepage et al., 2009). Since then, a number of studies have been investigating analogical learning for performing more specific machine translation tasks. Langlais et al. (2009) applied it to translating medical terms, and Langlais and Patry (2007) investigated the more specific task of translating unknown words, a problem simultaneously investigated in (Denoual, 2007). Recently, Langlais (2013) applied formal analogies to transliterate English proper names into Chinese.

Those works suggest, at least on the tasks investigated, that analogical translation typically shows better precision than phrase-based Statistical MT (SMT), but at a much lower recall. Still, the analogical devices tested in these works vary from task to task, making it difficult to draw a clear picture of the strengths and weaknesses of analogy-based translation. In this study, we perform a systematic comparison of an analogical and a phrase-based MT engine for the translation of fourteen different testbeds. We also improve the state-of-the-art of analogical learning by revisiting the aggregation step of the process. In particular, we observe that ranking analogical candidates according to random forests improves the performance of the analogical device, over training a classifier, as proposed for instance in (Langlais, 2013). On each task we tackle, we report improvements to the state-of-the-art in analogical learning.

In the remainder of this paper, we describe the principle of analogical learning and sketch our analogical device in Section 2. We describe our experimental protocol in Section 3. We analyze the performance of several variants of our analogical device in Section 4 and compare it to a state-of-the-art phrase-base SMT engine. We conclude this work and discuss future avenues in Section 5.

2 ANALOGICAL LEARNING

2.1 Principle

We note $[x : y :: z : ?]$ an *analogical equation*. It can have 0 or several solutions, depending on the definition of analogy being considered. We are given a training set (or memory) of pairs of *input* and *output* forms that are in (translation) relation: $L = \{\langle x_1, y_1\rangle, \ldots, \langle x_l, y_l\rangle\}$, and we note $\tau(x)$ the set of output forms to which the input form x corresponds in the training set: $\tau(x) = \{y : \langle x, y\rangle \in L\}$.

Given an input form u unseen at training time, analogical learning generates its associated output form (in our case its translation), by accomplishing 3 steps. First, analogies in the input space $[x : y :: z : u]$ are searched for. Second, output equations $[x' : y' :: z' : ?]$ are solved for all x', y', and z' in $\tau(x)$, $\tau(y)$, and $\tau(z)$ respectively. By applying those two steps (that we call the *generator*), a number of candidate solutions are typically produced. They need to be aggregated. This is the purpose of the third step, or *selector*. Note that for the mapping to happen between input and output strings, there is no attempt to align subsequences of forms in both spaces, as it is typically done in statistical MT. There is actually no alignment whatsoever: analogies are treated in each space separately, and the mapping is the result of the inductive bias which promotes that an analogy in the input space corresponds to an analogy in the output space.

Figure 1 depicts the overall process for the translation of the English term *proton pump inhibitors* into Spanish, given a memory of pairs such as $\langle$*blood coagulation factors*, *factores de coagulación sanguínea*$\rangle$ and $\langle$*proton pumps*, *bombas de protones*$\rangle$. 6 input analogies are being identified (2 are reported), therefore 6 (output) equations are being solved, yielding a total of 5268 different forms that are sorted in decreasing order of frequency with which they have been generated. This is the output of the generator. The reference translation (in bold) ranks 11$^{\text{th}}$ according to frequency. The aggregator finally selects two candidates from this list. The best ranked one according to the aggregator is the correct translation.

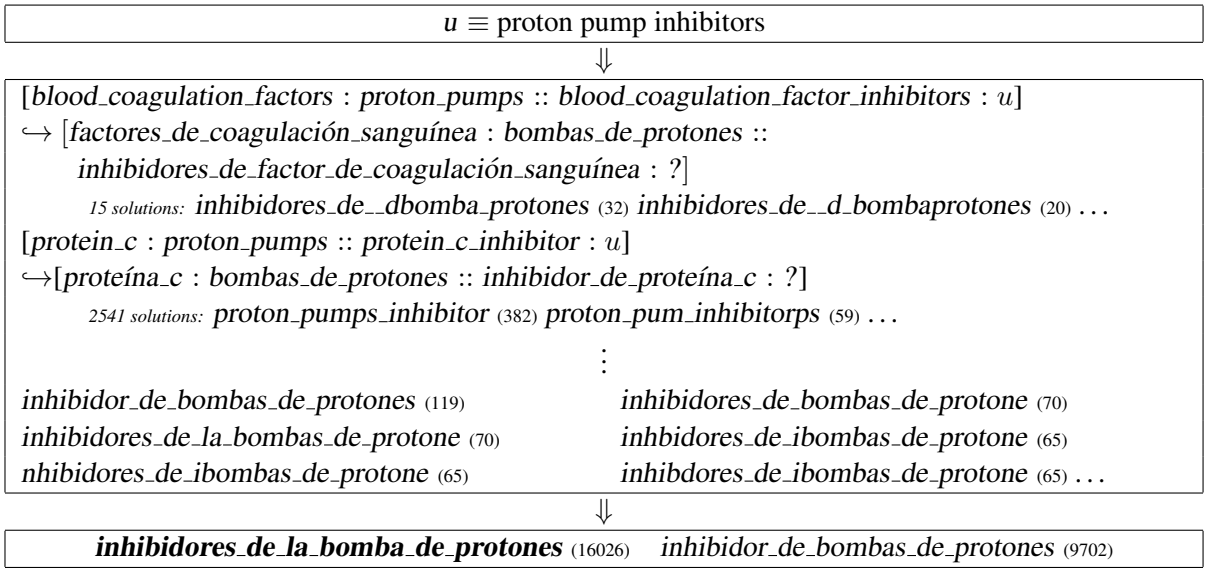

Figure 1: Excerpt of the translation session of the English term *proton pump inhibitors* into Spanish. The reference translation is in bold. Spaces are underlined for readability.

2.2 Implementation

Implementing such a learning procedure requires the definition of a formal analogy, the implementation of an analogical solver, as well as a way to handle computational issues: the identification of input analogies is an operation a priori cubic in the size of the input space. We describe each component of our implementation below. In practice, and for the tasks we consider in this work, our implementation allows the translation of an input form within a few seconds on average.

We would like to point out that analogical learning often suffers from a silence issue, that is, there are (input) forms for which no solution is provided. This may happen because no input analogy is identified, or because none yields an output equation with solutions. In contrast, there are many forms for which several candidate translations will be provided, thus the need for a good aggregator (see next section). This happens because an equation typically allows many solutions, and because many input analogies might be identified for solving a given input form.

Formal Analogy We used the definition of formal analogy proposed by Yvon et al. (2004), where an analogy is defined in terms of *d-factorizations*. A d-factorization of a string x over an alphabet Σ, noted f_X, is a sequence of d *factors* $f_X \equiv (f_X^1, \ldots, f_X^d)$, where $f_X^i \in \Sigma^*$ for all i, and such that $f_X^1 \odot f_X^2 \odot f_X^d \equiv x$; where $\odot$ denotes the concatenation operator.

Definition 1. *$\forall\, x, y, z$ and $t \in \Sigma^\star$, [x : y :: z : t] **iff** there exists a 4-uple of d-factorizations (f_X, f_Y, f_Z, f_t) of x, y, z and t respectively, 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 holds is called the degree of the analogy.*

For instance, [*protein_c* : *proton_pumps* :: *protein_c_inhibitor* : *proton_pump_inhibitors*] because of the 4-uple of 3-factorizations shown in Fig. 2, whose factors are aligned column-wise for clarity, and where spaces (underlined) are treated as regular characters. There is no 4-uple of d-factorizations, with d smaller than 3. Therefore, the degree of this analogy is 3. Note that there are many 4-uple of d-factorizations for d greater than 3.

Figure 2: A 4-uple of 3-factorizations demonstrating that [*protein_c* : *proton_pumps* :: *protein_c_inhibitor* : *proton_pump_inhibitors*].

f_X	$\equiv$	(	*protein_c*	ϵ	ϵ)
f_Y	$\equiv$	(	*proton_pump*	ϵ	*s*)
f_Z	$\equiv$	(	*protein_c*	*_inhibitor*	ϵ)
f_t	$\equiv$	(	*proton pump*	*_inhibitor*	*s*)

Analogical Solver With the aforementioned definition, it has been showed by Yvon et al. (2004) that the set of solutions to an analogical equation is a rational language, therefore we can build a finite-state machine for encoding those solutions. In practice however, the automaton is non-deterministic, and in the worst case, enumerating the solutions can be exponential in the length of the forms involved in the equation. We adopted the solution proposed in (Langlais et al., 2009) which consists in sampling this automaton without building it. The more we sample this automaton, the more solutions we produce. It is sufficient to note that typically, a solver produces several solutions to an equation, many being simply spurious, which means that, while they obey the definition of formal analogy, they are not valid forms.

Figure 3: Three most frequent solutions to the equation [*protein_c* : *proton_pumps* :: *protein_c_inhibitor* : *?*] along with their frequency, as a function of the number of samples considered 10^n. nb stands for the total number of solutions produced.

n	nb	solutions		
1	43	*p_inhibitorroton_pumps* (2)	*proton_p_inhiubitomrps* (2)	*prot_ion_pnhibitumorps* (2)
2	320	*proton_pumps_inhibitor* (8)	*proton_pum_inhibitposr* (4)	*prot_inhibion_pumtorps* (4)
3	2 597	*proton_pumps_inhibitor* (121)	*roton_pumpps_inhibitor* (19)	*proton_pump_inhsibitor* (19)
4	16 006	*proton_pumps_inhibitor* (764)	*proton_pump_inhibsitor* (103)	*proton_pump_isnhibitor* (95)
5	72 610	*proton_pumps_inhibitor* (3706)	*proton_pump_sinhibitor* (501)	*proton_pump_inhibitosr* (481)

To illustrate this, Figure 3 reports the solutions produced to the equation [*protein_c* : *proton_pumps* :: *protein_c_inhibitor* : *?*] by our implementation of the solver, as a function of the number of samplings done in the automaton. Clearly, many solutions are not valid forms in English, although they define

proper solutions according to the aforementioned definition. Note that with enough sampling, the solution *proton_pumps_inhibitor* (involving a degree-2 analogy) is the most frequently generated one, while the solution *proton_pump_inhibitors* involved in the analogy illustrated in Figure 2 is generated less often (typically at the 10^{th} position).

Searching input analogies Identifying input analogies for an input term u is an operation a priori cubic in the size of the input space. Langlais and Yvon (2008) developed an algorithm for speeding up the search procedure that we adopted in this work. The main idea is to exploit a property of formal analogies (Lepage and Shin-ichi, 1996):

$$[x : y :: z : u] \Rightarrow |x|_c + |u|_c = |y|_c + |z|_c \quad \forall c \in \mathcal{A} \tag{1}$$

where $\mathcal{A}$ is the (input) alphabet, and $|x|_c$ stands for the number of occurrences of symbol c in x.

The strategy consists in first selecting a form x in the input space. This enforces a set of necessary constraints on the counts of symbols that any two forms y and z must satisfy for $[x : y :: z : u]$ to hold. By considering all forms x in turn, we collect a set of candidate triplets for u. We then have to find out which of these triplets form an analogy with u. Formally, we search for:

$$\begin{aligned}\{\langle x, y, z\rangle : \quad & x \in \mathcal{I}, \\ & \langle x, y\rangle : y \in \mathcal{I} \text{ and } |x|_c + |u|_c = |y|_c + |z|_c \quad \forall c \in \mathcal{A}, \\ & [x : y :: z : u]\}\end{aligned} \tag{2}$$

where $\mathcal{I} \equiv \{x_1, \ldots, x_l\}$. This strategy relies on the fact that one can efficiently identify the pairs $\langle y, z\rangle$ that satisfy a set of constraints on symbol counts. See (Langlais et al., 2009) for the *tree-count* solution we implemented in this work.

3 Experimental Protocol

3.1 Tasks

We use two families of tasks in this study. The first one concerns the translation of medical terms, the second one is about transliterating proper names. The main characteristics of the datasets we consider are reported in Table 1. If both tasks are of importance in practice, we admit that they are rather specific. The reason for this is that analogical learning is quite computationally intensive. Therefore, tackling broader tasks, such as those typically considered in MT evaluation campaigns is currently too challenging.

Medical term translation We use the datasets described in (Langlais et al., 2009). Part of the data 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. The MESH material concerns 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). The material was split in three randomly selected parts (TRAIN, DEV and TEST), so that the development and test material contain exactly 1000 terms each. Roughly a third of the examples are pairs of single-word terms.

For the Spanish-English language pair, a set of medical terms from the Medical Drug Regulatory Activities thesaurus (MEDDRA) is also available. This dataset contains roughly three times more terms than the Spanish-English material from the MESH dataset. Forms in the dataset are typically longer and the percentage of examples that are pairs of single-word terms is only 5.6%. This set is used for studying how the silence rate of analogical learning evolves with the size of the training set.

We are pleased to share those datasets. They can be downloaded at `http://rali.iro.umontreal.ca/rali/?q=en/12-medical-translation-tasks`.

Proper name transliteration This task is part of the NEWS evaluation campaign conducted in 2009 (Li et al., 2009). The organizers of this evaluation campaign kindly provided us with the Chinese-English dataset. This task has been investigated recently by Langlais (2013). This allows a direct comparison of our analogical system. We also consider the reverse transliteration direction, *i.e.*, the transliteration of

Chinese proper names into English. This was done by simply switching the source and target languages in the NEWS dataset.

	TRAIN		TEST	DEV	
	nb	$avg.$	nb	oov%	oov%	
MESH						examples:
FI	19 787	19.3	1 000	65.0	63.8	*orthodontic retainers*
FR	17 230	21.5	1 000	35.8	36.8	↪ FI:*tandregleringshjälpmedel, förankrade*
RU	21 407	38.5	1 000	42.3	45.1	
ES	19 021	21.5	1 000	37.4	34.9	*aid to families with dependent children*
SW	17 090	17.3	1 000	69.3	70.0	↪ SW:*bidrag till barnfamiljer*
MEDDRA						*poor urinary stream*
ES	65 276	34.6	1 000	7.1	7.1	↪ ES: *chorro de orina débil*
NEWS						*Abberley* → CN:阿伯利
CN	31 961	9.5	2 896	—	—	*Schemansky* → CN:谢曼斯基

Table 1: Main characteristics of our datasets. nb indicates the number of pairs of terms in a bitext, $avg.$ indicates the average length (in symbols) of the foreign forms; oov% indicates the percentage of out-of-vocabulary types (space-separated types of TEST or DEV unseen in TRAIN).

3.2 Evaluation Metrics

All the tasks we consider are characterized by a rather high out-of-vocabulary rate (see Table 1). Thus, word-based translation is not an adequate solution. Therefore, we devised engines which translate sequences of symbols (characters), without taking into account the notion of word.[1] In particular, spaces in forms were considered as any ordinary symbol. Measuring how close a candidate translation is to a reference is of little interest here, since typically, a medical term only has one reference translation that we seek to discover. Therefore, rewarding partially correct translations (like a metric such as BLEU (Papineni et al., 2002) does) is not especially useful. Therefore we report the *accuracy* of the first candidate proposed by a translation device for each source term. Accuracy is measured as the percentage of test forms for which the first candidate is the sanctioned one. So in the example of Figure 1, the aggregator illustrated in the bottom frame would get one point since the first translation produced is the sanctioned one, while an aggregator that would pick the most frequently generated candidate would receive no point. Accuracy is the main metric of the NEWS evaluation campaign, and we used the NEWS 2009 official evaluation script[2] in order to compute it. Also of interest for the analogical devices, is the *silence rate*, computed as the percentage of input forms for which no output is generated. As we will see, on some tasks, this ratio can be rather high, a clear limitation of the analogical approach we discuss in Section 5.

3.3 Systems

3.3.1 Reference System

We compare a number of analogical devices to the state-of-the-art statistical translation engine Moses (Koehn et al., 2007). In a nutshell, SMT seeks to find the optimal translation $\hat{e}$ of a sentence f using to a log-linear combination of models (h_i), including a language model $p(e)$ which scores how likely a hypothesis is in the target language, and a translation model $p(f|e)$ which predicts the likelihood that two sentences are translations:

$$\hat{e} = \underset{e}{\operatorname{argmax}}\, p(f|e)p(e) \approx \underset{e}{\operatorname{argmax}} \exp\left(\sum_i \lambda_i h_i(e, f)\right) \quad (3)$$

[1] We tried it, but the results are very low.

[2] http://translit.i2r.a-star.edu.sg/news2009/

We trained such a system at the character level,[3] very similarly to the approach described in (Finch and Sumita, 2010). Such a system has been massively used as a key component by the participants of the NEWS 2009 evaluation campaign. We used the default configuration of Moses for training and testing the SMT engine. We trained a 5-gram character-based language model on the target part of the TRAIN material.[4] We used the DEV corpus for tuning the coefficients (λ_i) given to each model. The resulting system have high BLEU scores (*e.g.*, 55.7 for the CN–EN NEWS task). A random extract of the phrase-table learnt by Moses for the English-Swedish system is shown in Figure 4.

Figure 4: Phrases stored in the SW–EN phrase-table, along with 4 estimations of their likelihood

eckos	echos	*0.303 0.006*	*0.303 0.002*
, _ kvinn	, _ fema	*0.101 8.3e-09*	*0.303 2.5e-11*
eckrina	eccrine	*0.151 0.009*	*0.303 0.001*
edel	ator	*0.002 4.6e-06*	*0.002 1.9e-06*

3.3.2 Analogical Systems

We ran our analogical generator for translating the DEV set, using the TRAIN set as a memory. The candidate translations generated were used for training our aggregators in a supervised way. Then, we generated the translation of the TEST terms with our analogical device, making use of the TRAIN and the DEV set as a memory. Adding the DEV corpus to the memory used by the generator is acceptable since it does not involve training. We only consider the (at most) 100 most frequently generated forms for each input term. This certainly decreases the recall of the analogical device, but simplifies the overall process. These candidates are passed on to the aggregator, and one candidate is finally selected.

Aggregators A number of aggregators have been proposed in the literature. Lepage and Denoual (2005b; Stroppa and Yvon (2005) keep the candidate that has been generated the most frequently. We call this aggregator FREQ henceforth. Langlais et al. (2009) trained a binary classifier to recognize good examples from bad ones. A training instance in their case was constituted by an input analogy, and the corresponding output equation along with one solution produced. Therefore, for the translation of the input form u, any pair ($[x : y :: z : u]$, $[x' : y' :: z' : c]$), with x', y', and z' in $\tau(x')$, $\tau(y')$, and $\tau(z')$ respectively, and c a candidate translation would be considered for classification. The authors had to face a particularly unbalanced classification task. Indeed, when translating a test form, a large number of input analogies can be considered (hundreds) and therefore a large number of output equations, each generating potentially numerous solutions (recall the translation session in Figure 1). They reported for instance that on the English-to-Finnish translation direction, they had over 2.7 million instances to classify among which slightly less than 4200 were positive ones. Not only is this task very unbalanced, it is also challenging to train a classifier on that many instances.

In this work, we reframe the classification task as one of identifying the correct candidate among the 100 most frequently generated ones. An instance in this setting is simply a candidate form, and not a pair of analogies as in (Langlais et al., 2009). This is still an unbalanced task, since typically at most one candidate will be correct, but the ratio 1:100 is more manageable, and the classification task is easier to deploy. A total of 81 features are computed for each candidate form:

ANA is a set of 59 features (mostly analogical ones, therefore the name). Some features are characterizing the candidate solution thanks to a character-based language model (the same 5-gram language model used by Moses). Others are characterizing the process with which a given candidate is generated, such as the number of input analogies involved, the number of target equations that generated the candidate, the average degree of the analogies involved, etc. The remaining features are cohort-based ones, such as the rank of the candidate according to frequency, to the language model, etc.

[3] This was done by separating each character in the training material by a space; true spaces being previously substituted by a special character not belonging to the alphabet.

[4] A Markov model of order 4. We tried higher order models, without gains.

IBM is a set of 18 features that are capitalizing on statistical word alignment. The alignment models being used are word-based generative models that are exploited by Moses in order to build the phrase table, namely IBM models, therefore the name of the feature set. Different likelihood-based features were computed, as well as rank features (the rank of the likelihood of the candidate in the cohort of candidates, the ratio of its likelihood over the highest likelihood in the cohort, etc.). To our knowledge, this is the first attempt to capitalize on such features in the analogical sphere.

MOS is a set of 4 features that are exploiting the n-best solutions we asked Moses to produce. The idea being that if Moses ranks a given analogical candidate well (in rank or in score), this is a good indicator of the salience of this candidate. The two main features are the rank of the candidate in the n-best list and its score as given by Moses (or 0 if Moses does not produce the candidate).

We point out that an analogical device with an aggregator that uses the features ANA and IBM is basically making use of the same models (language and IBM) as those used by Moses. It is therefore interesting to compare this configuration to Moses. Also, the aggregators that are making use of the MOS features are performing a kind of combination that has not been explored so far. Note also that we did not engineer task-specific features. For instance, for the medical term translation task, terms and their translation often share the same latin root, which could be exploited to boost performance.

We investigated two families of classifiers: voted-perceptrons (Freund and Schapire, 1999) and support vector machines (Cortes and Vapnik, 1995). We investigated all the metaparameters that `LibSVM` (Chang and Lin, 2011) offers (penalization, kernels, etc.), but did not manage to outperform the performance of the former classifier (an in-house implementation) that we trained with 500 epochs. Therefore we only report the results of the voted-perceptron classifier (VP). Classifying each candidate solution separately is not optimal. This is why we also investigated reranking algorithms in this study. To our knowledge, this is the first time reranking is applied in analogical learning. We tested the algorithms implemented in `RankLib`[5] and `SVMRank`[6] toolkits, and found random forests (Breiman, 2001) to be the most beneficial. We note it RF in the sequel. We only considered bipartite ranking in this work (Argarwal, 2005).

4 Results

4.1 MESH

The accuracy of the translation devices we trained are summarized in Table 2 for the 10 translation directions we tested. This table calls for several comments. First, it is noticeable that our implementation of analogical learning with the FREQ aggregator (line LYZ) outperforms the equivalent configuration in (Langlais et al., 2009) by roughly 10 absolute points in accuracy. We also observe a slight reduction of the silence rate, which still remains high, since on average 54.6% of the test forms do not receive any candidate solution. Second, we observe that Moses slightly outperforms the FREQ variant at a silence rate of 0 (a decision is always returned by Moses). This suggests that FREQ is actually more precise than Moses and calls for a simple combination where the analogical device is trusted whenever it produces a candidate solution, and Moses otherwise. This is illustrated in line CASC(FREQ,MOSES). We observe a clear improvement over each system: almost 10 absolute accuracy points on average are gained by this combination (38.6%). Third, we observe that the aggregators that are relying on a classifier or a reranker offer better performance than picking the most frequently generated form (as done by FREQ). The gains are not especially high, but are consistent over all translation directions. Overall, it seems that the random forest reranker we investigated (the best reranker we tried) offers the best performance on average. This represents 92% of the reachable accuracy according to line ORACLE which involves a perfect classifier. This validates the usefulness of the features we designed. As far as features are concerned, it seems that using all of them leads to better performance overall, and that the configurations that are making use of the ANA and IBM feature sets are comparable or higher than Moses. Cascading the best analogical device with Moses (last line) finally gives a slight boost in accuracy. In the end, the best system we tested correctly translated 41.9% of the test terms in the first position on average across translation directions.

[5] `http://people.cs.umass.edu/~vdang/ranklib.html`

[6] `http://www.cs.cornell.edu/people/tj/svm_light/svm_rank.html`

	→ EN					EN →					
	FR	RU	FI	ES	SW	FR	RU	FI	ES	SW	*avg.*
LYZ	18.1	20.8	16.4	20.3	18.2	14.6	18.7	14.9	19.5	15.4	17.7
	(61.5)	*(57.9)*	*(55.2)*	*(57.4)*	*(55.4)*	*(58.8)*	*(53.8)*	*(52.9)*	*(53.0)*	*(57.2)*	*(56.3)*
FREQ	27.3	29.1	28.5	30.5	28.3	21.8	29.0	24.7	29.8	26.3	27.5
	(59.3)	*(56.7)*	*(53.7)*	*(55.6)*	*(54.3)*	*(56.0)*	*(52.5)*	*(50.9)*	*(51.6)*	*(55.2)*	*(54.6)*
MOSES	22.3	33.4	27.0	29.0	38.8	20.0	30.5	26.4	28.6	37.0	29.3
VP(ANA)	28.4	29.8	29.8	31.9	29.7	23.2	31.0	27.2	32.3	27.9	29.1
VP(ANA+IBM)	28.8	31.8	31.6	32.4	31.2	24.5	32.3	28.4	34.2	29.2	30.4
VP(ANA+IBM+MOS) †	29.2	32.3	31.6	32.8	31.9	25.0	32.6	28.8	34.0	30.1	30.8
RF(ANA)	28.3	29.8	30.7	32.0	29.5	23.0	31.2	27.4	31.6	28.3	29.2
RF(ANA+IBM)	29.1	31.6	31.8	32.8	31.0	24.4	32.4	28.7	33.5	30.1	30.5
RF(ANA+IBM+MOS)	29.4	31.8	32.2	32.9	32.4	24.9	32.5	29.9	34.0	31.1	31.1
ORACLE	31.3	34.0	34.9	35.2	34.9	28.2	35.7	33.2	37.3	33.3	33.8
	(68.7)	*(66.0)*	*(65.1)*	*(64.8)*	*(65.1)*	*(71.8)*	*(64.3)*	*(66.8)*	*(62.7)*	*(66.7)*	*(66.2)*
casc(FREQ,MOSES)	36.9	42.4	37.7	41.6	43.8	29.6	38.9	34.3	40.7	39.9	38.6
casc(†,MOSES)	38.8	45.6	40.8	43.9	47.4	32.8	42.5	38.4	44.9	43.7	41.9

Table 2: Accuracy on the MESH tasks. Figures in parenthesis are silence rates. LYZ stands for the system described in (Langlais et al., 2009), reproduced according to Table 4, p. 492. *avg.* indicates the average over the 10 translation directions.

4.2 MEDDRA

The results presented so far show that the analogical device is more accurate than the statistical one, but that it suffers from a high silence rate. We tested whether increasing the size of the training set would lower the silence rate. We used the datasets of MEDDRA for this. The results are reported in the left column of Table 3. We observe that the silence rate decreases drastically, since less than a fourth of the test forms do not receive a candidate translation. We also observe that the analogical devices, even the simplest FREQ, are far more accurate than Moses (over 30 absolute points on average). The poor performance of the SMT engine might be explained by the fact that the forms in the MEDDRA datasets are longer in terms of characters, therefore reducing the chance of getting the full translation right. Again, combining both approaches does improve accuracy, but the improvement is small since Moses is much less accurate on this task. Also, we observe that using a classifier is preferable to picking the most frequently generated form, and again, the random forest reranker delivers the best performance on average. It is noticeable however, that the performance is far less than the oracle's, therefore, there is still room for improvement.

4.3 NEWS

The right column of Table 3 summarizes the performance of the transliteration devices we trained on the NEWS tasks. The silence rate is rather low (less than 4%). Here again, we observe that aggregating by classifying or reranking is preferable to picking the most frequent solution. There is no clear difference between random forest and voted perceptron here. On the English-to-Chinese transliteration tasks, Moses outperforms the analogical devices, but the opposite is observed for the reverse transliteration direction. Our best configuration slightly outperforms the best analogical device reported in (Langlais, 2013), but the gain is likely not significant.

	MEDDRA				NEWS			
	ES–EN		EN–ES		CN–EN		EN–CN	
FREQ	52.2	*(25.1)*	45.5	*(16.7)*	17.2	*(2.5)*	43.3	*(3.7)*
MOSES	10.2		11.0		15.4		66.6	
VP(ANA)	55.1		46.8		20.0		57.3	
VP(ANA+IBM)	56.2		46.9		20.9		59.5	
VP(ANA+IBM+MOS) †					21.4		64.2	
RF(ANA)	54.1		49.3		20.9		57.8	
RF(ANA+IBM)	55.7		49.5		21.6		59.2	
RF(ANA+IBM+MOS)					22.3		64.1	
ORACLE	64.3	*(34.4)*	61.8	*(38.2)*	64.9	*(32.9)*	81.5	*(18.5)*
casc(FREQ,MOSES)	53.2		46.7		17.5		44.9	
casc(†,MOSES)	—		—				68.9	
(Langlais, 2013)							68.5	

Table 3: Accuracy on the MEDDRA and NEWS tasks. The performance of (Langlais, 2013) is taken from Table 1 p. 687.

4.4 Examples of translations

We conducted a random inspection of the outputs produced by Moses and by the analogical device which uses a voted perceptron classifier trained on the ANA and the IBM features.[7] We report in Figure 5 a few examples that we found representative of the problems each translation device faces. The FI–EN example shows a case where Moses fails to produce a valid sequence of words. The EN–ES example illustrates the weakness of Moses at reordering words. The CN–EN example shows the incorrect transliterations made by both systems, and the EN–CN one illustrates a failure of the analogical engine were *ph* and *us* are transliterated separately.

MESH(FI–EN)	*hammasytimen sairaudet*	NEWS(CN–EN)	本尼迪克特
Analog	*dental marrow diseases*	Analog	Bennidickt
MOSES	*dental ne diseases*	MOSES	BenniDickert
Reference	*dental pulp diseases*	Reference	Benedict
MEDDRA(EN–ES)	*intrinsic asthma with status asthmaticus*	NEWS(EN–CN)	Adolphus
Analog	*asma intrínseca con estatus asmático*	Analog	阿道夫厄斯
MOSES	*intrínseco asmático con estatus asmático*	MOSES	阿道弗斯
Reference	*asma intrínseca con estatus asmático*	Reference	阿道弗斯

Figure 5: Examples of analogical and phrase-based outputs

5 Discussion

We have applied analogical learning on a number of key tasks involving various language pairs. Overall, we confirm the findings of Langlais et al. (2009) and Langlais (2013) that analogical devices are typically more accurate than statistical phrase-based SMT, but that they are too often silent. We also verified that cascading the analogical device with Moses increases accuracy. We compared a number of classification algorithms and rerankers, and observed that overall, reranking by random forest

[7] This variant fares well compared to Moses in terms of information used (same language and IBM models).

delivers the best performance. Our implementation outperforms previously reported ones. Our generator is more efficient than the one described in (Langlais et al., 2009). Reranking candidate solutions is preferable to their classification, as proposed in (Langlais, 2013). In order to foster reproducibility, the datasets related to the medical-translation tasks we investigated can be downloaded at `http://rali.iro.umontreal.ca/rali/?q=en/12-medical-translation-tasks`.

We believe this systematic comparison shows the high potential of analogical learning as a translation engine. Still, this work raises a number of issues that we must address. First, we need to find ways to remedy analogical learning's high silence rate. Lepage and Denoual (2005b) describe a recursive process where the input form is split into two parts whenever no solution is returned in the first place. This process is at the very least costly and deserves further investigations. Lepage and Lardilleux (2008) augments the training set with sub-sentential alignment (bootstrapping). Second, the solver we use is producing many solutions that are currently ranked according to frequency. We are addressing the issue of producing less, but more accurate solutions, by integrating structured learning in the solver. Last, we investigated here the translation of sequences of characters on modestly sized tasks. We want to tackle broader translation tasks, *e.g.*, translating plain sentences, as done in (Lepage and Denoual, 2005b), to see whether our analogical device is still beneficial.

Acknowledgements

We thank the reviewers for their valuable comments and apologize for having failed to taking all of them into account in this version. This work has been partially funded by the Natural Science and Engineering Research Council of Canada. We are grateful to Fabrizio Gotti for his advice.

References

Shivani Argarwal. 2005. A study of the bipartite ranking problem in Machine Learning. Technical report, University of Illinois.

Leo Breiman. 2001. Random Forests. *Machine Learning*, 45(1):5–32.

Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A Library for Support Vector Machines. *ACM Trans. Intell. Syst. Technol.*, 2(3):1–27, May.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine Learning*, 20(3):273–297.

Étienne Denoual. 2007. Analogical translation of unknown words in a statistical machine translation framework. In *MT Summit XI*, pages 135–141, Copenhagen, Denmark.

Andrew Finch and Eiichiro Sumita. 2010. Transliteration Using a Phrase-based Statistical Machine Translation System to Re-score the Output of a Joint Multigram Model. In *2nd Named Entities Workshop (NEWS'10)*, pages 48–52, Uppsala, Sweden.

Yoav Freund and Robert Schapire. 1999. Large Margin Classification Using the Perceptron Algorithm. *Mach. Learn.*, 37(3):277–296.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *45th ACL*, pages 177–180. Interactive Poster and Demonstration Sessions.

Philippe Langlais and Alexandre Patry. 2007. Translating Unknown Words by Analogical Learning. In *EMNLP*, pages 877–886, Prague, Czech Republic.

Philippe Langlais and François Yvon. 2008. Scaling up Analogical Learning. In *22nd International Conference on Computational Linguistics (COLING 2008), Poster session*, pages 51–54, Manchester, United Kingdom, Aug.

Philippe Langlais, François Yvon, and Pierre Zweigenbaum. 2009. Improvements in Analogical Learning: Application to Translating multi-Terms of the Medical Domain. In *12th EACL*, pages 487–495, Athens.

Philippe Langlais. 2013. Mapping Source to Target Strings without Alignment by Analogical Learning: A Case Study with Transliteration. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 684–689, Sofia, Bulgaria.

Yves Lepage and Étienne Denoual. 2005a. Aleph: an EBMT system based on the preservation of proportional analogies. In *2nd IWSLT*, pages 47–54, Pittsburgh, USA.

Yves Lepage and Étienne Denoual. 2005b. Purest ever example-based machine translation: Detailed presentation and assesment. *Mach. Translat*, 19:25–252.

Yves Lepage and Adrien Lardilleux. 2008. The GREYC Translation Memory for the IWSLT 2007 Evaluation Campaign. In *4th IWSLT*, pages 49–54, Trento, Italy.

Yves Lepage and Ando Shin-ichi. 1996. Saussurian Analogy: A Theoretical Account and Its Application. In *7th COLING*, pages 717–722.

Yves Lepage, Adrien Lardilleux, Julien Gosme, and Jean-Luc Manguin. 2008. The GREYC Translation Memory for the IWSLT 2008 Evaluation Campaign. In *5th IWSLT*, pages 39–45, Hawai, USA.

Yves Lepage, Adrien Lardilleux, and Julien Gosme. 2009. The GREYC Translation Memory for the IWSLT 2009 Evaluation Campaign: one step beyond translation memory. In *6th IWSLT*, pages 45–49, Tokyo, Japan.

Haizhou Li, A. Kumaran, Vladimir Pervouchine, and Min Zhang. 2009. Report of NEWS 2009 machine transliteration shared task. In *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration*, NEWS '09, pages 1–18.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proc. of the 40th ACL*, pages 311–318, Philadelphia, Pennsylvania, USA.

Nicolas Stroppa and François Yvon. 2005. An Analogical Learner for Morphological Analysis. In *9th Conf. on Computational Natural Language Learning (CoNLL)*, pages 120–127, Ann Arbor, USA.

François Yvon, Nicolas Stroppa, Arnaud Delhay, and Laurent Miclet. 2004. Solving Analogies on Words. Technical Report D005, École Nationale Supérieure des Télécommuncations, Paris, France.