An analogical learning approach to translating terms

Traduction de termes à base d'analogies

Philippe Langlais
François Yvon
Pierre Zweigenbaum

2008D013

Octobre 2008

Département Informatique et Réseaux
Groupe IC2 : Interaction, Cognition et Complexité

An Analogical Learning Approach to Translating Terms

Traduction de termes à base d'analogies

Philippe Langlais
DIRO, University of Montreal
C.P. 6128 Suc. Centre-Ville
H3C 3J7 Montreal, Canada
felipe@iro.umontreal.ca

François Yvon and **Pierre Zweigenbaum**
Université Paris Sud 11 & LIMSI, CNRS
B.P. 133
91403 Orsay Cedex, France
{yvon,pz}@limsi.fr

Résumé

La gestion terminologique occupe une part importante de l'activité de traduction. Des outils de gestion de bases terminologiques permettent d'assister le terminologue ou le traducteur lorsque les traductions des termes d'intérêt sont répertoriées. Trouver la traduction d'un nouveau terme reste cependant une tâche délicate pour laquelle peu d'outils sont adaptés. Dans cette étude, nous proposons de traduire les termes inconnus par *apprentissage analogique*. Nous montrons sur une base de termes simples du domaine médical, qu'une variante de notre approche permet de produire des traductions (au plus 9) pour 64% des termes de notre jeu de test, et que la traduction de référence y est présente dans 80% des cas.

Abstract

Handling terminology is an important matter in a translation workflow. However, current Machine Translation (MT) systems do yet not propose anything proactive upon tools which assist in managing terminological databases. In this work, we propose an original approach to translating terms that combines analogical learning (for proposing candidate terms) and more conventional statistical classification tools (to validate the hypothesized candidates). Applied to translating medical terms, we show that 64% of unknown terms receive at least one translation (at most 9), of which one is correct 80% of the time.

1 Introduction

If machine translation is to meet professional needs, it must offer a sensible approach to translating terms. Langlais and Carl (2004) measured that domain specific texts they considered contained a large proportion of sentences (more than 35%) with unknown terms; which ruined out the intelligibility of the translations produced by the generic (statistical) MT system they used.

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).

In this study, we are interested in automatically translating terms. More precisely, we are investigating to what extend, we can benefit from existing pairs of source-target terms in order to translate new terms. Such an approach would for instance find applications in assisting terminologist to enrich dedicated terminological databases or alternatively could simply be embedded in a translation engine.

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 amount of parallel data is 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 kettle of fish (see Section 5.4 for an illustration of that).

One might argue that domain-specific comparable (or perhaps unrelated) corpora are easier to acquire, in which case context-vector techniques such as the ones described in (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 do simply not exist.

Langlais and Patry (2007) described a work where they translate unknown words and phrases by analogical learning (Stroppa and Yvon, 2005). Their approach consists in identifying proportional analogies between words (or phrases) belonging to a seed bilingual lexicon. A similar idea has been independently investigated by Denoual (2007). In this work, we investigate whether analogical learning can carry over the task of translating single terms. We designed a system, AnaTerm, which extends the approach described by Langlais and Patry (2007) in several ways, notably by the introduction of a binary classifier trained to recognize valid analogies from spurious ones. The main novelty here is to combine the benefits of symbolic (analogical) learning in a candidate generation phase, and of statistical learning in a candidate selection step.

Although our system is not specialized for that purpose, we tested its performance on a task of translating terms from the medical domain. Our system could translate 64% of a set of unknown terms, producing a good translation in a list of at most 9 candidates 80% of the time. Since AnaTerm makes only use of a (small) domain-specific seed-lexicon, we believe it suits well the need of terminologists that usually have at their disposal an incomplete terminology for a given domain.

The paper is organized as follows. We recap in Section 2 the principle of analogical learning and describe how to apply it to our task. We describe our system in Section 3. We present our experimental protocol in Section 4 and evaluate AnaTerm in Section 5. In Section 6, we compare our approach to recently published alternatives. We finally discuss our work and present future avenues in Section 7.

2 Analogical Learning

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 $[\textit{believer} : \textit{unbelievable} = \textit{dreamer} : \textit{undreamable}]$.

Formal analogies can be defined in terms of factorizations (Stroppa and Yvon, 2005). Let x be a string over a finite alphabet Σ, a *factorization* of x, noted f_X, is a sequence of n *factors* $f_X = (f_X^1, \dots, f_X^n)$, such that $x = f_X^1 \odot f_X^2 \odot f_X^n$, where $\odot$ denotes the concatenation operation. We thus define:
$\forall (x, y, z, t) \in {\Sigma^\star}^4$, $[x : y = z : t]$ if and only if there exists factorizations $(f_X, f_Y, f_Z, f_t) \in ({\Sigma^\star}^d)^4$ of (x, y, z, t) such that, $\forall i \in [1, d]$, $(f_Y^i, f_Z^i) \in \{(f_X^i, f_t^i), (f_t^i, f_X^i)\}$. 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 this definition captures the examplar analogy introduced above, based on the following set of factorizations:

$$\begin{array}{lcl} f_X & \equiv & (\epsilon, \textit{believ}, \textit{er}) \\ f_Y & \equiv & (\textit{un}, \textit{believ}, \textit{able}) \\ f_Z & \equiv & (\epsilon, \textit{dream}, \textit{er}) \\ f_t & \equiv & (\textit{un}, \textit{dream}, \textit{able}) \end{array}$$

There is no smaller factorization in terms of the number of factors involved, and therefore, the degree of this (formal) analogy is 3. Note that the factors do not have to be linguistically sensible units.

In the sequel, we introduce the concept of *cofactor* of a formal analogy $[x : y = z : t]$ to be a vector of d alternations $[\langle f, g \rangle_i]_{i \in [1,d]}$ where an *alternation* is defined formally as:

$$\langle f, g \rangle_i = \begin{cases} (f_X^{(i)}, f_Z^{(i)}) & \text{if } f_X^{(i)} \equiv f_Y^{(i)} \\ (f_Y^{(i)}, f_Z^{(i)}) & \text{otherwise} \end{cases}$$

The cofactors of the examplar analogy are: $[(\epsilon, \textit{un}), (\textit{believ}, \textit{dream}), (\textit{er}, \textit{able})]$.

We call an *analogical equation* an analogy where one item (usually the forth one) is missing and we note it $[x : y = z : ?]$.

2.1 Analogical Inference

Analogical learning belongs to the family of lazy learning techniques (Aha, 1997). Let $\mathcal{L} = \{(i,o) \,|\, i \in \mathcal{I}, o \in \mathcal{O}\}$ be a set of observations, where $\mathcal{I}$ (resp. $\mathcal{O}$) is the set of possible forms of the input (resp. output) linguistic system of the application. We denote $I(u)$ (resp. $O(u)$) the projection of u into the input (resp. output) space; that is, if $u = (i,o)$, then $I(u) \equiv i$ and $O(u) \equiv o$. In this setting, training simply consists in memorizing the associations between input and output that are observed in $\mathcal{L}$. For an incomplete observation $u = (i, ?)$, the inference procedure is a three step process :

1. build $\mathcal{E}_{\mathcal{I}}(u) = \{\langle x, y, z\rangle \in \mathcal{L}^3 \mid [I(x) : I(y) = I(z) : I(u)]\}$, the set of input triplets that define an analogy with $I(u)$.
2. build $\mathcal{E}_{\mathcal{O}}(u) = \{o \in \mathcal{O} \mid \exists\langle x, y, z\rangle \in \mathcal{E}_{\mathcal{I}}(u)$ 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. select candidates among $\mathcal{E}_{\mathcal{O}}(u)$.

To give one example, assume $\mathcal{L}$ contains the following entries :[1] *(carpine,caprin)*, *(actine,actin)*, *(apraxie,apraxia)*. We might translate the French term *ataxie* into the English term *ataxia* by:

1. identication of the input (French) triplet: $\langle$*caprine, actine, apraxie*$\rangle$;

2. projection of this triplet onto the output (English) space, yielding the equation [*caprin* : *actin* = *apraxia* : ?], and resolution of this equation.

3. selection, amongst the set of solution, of *ataxia*, which is one of the solution identified in step 2.

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 analogy in the output space).

This general setting can be applied to the task we are interested in, that is, translating new terms. The training corpus is in our case a set of pairs of source-target terms. The input (resp. output) space is the set of all the possible source (resp. target) terms of a domain.

3 The `AnaTerm` system

`AnaTerm`, the term translation system designed according to the general principles detailled above, is composed of three modules: the *solver* which solves analogical equations; the *generator*, which encompasses the two first steps of analogical learning; and the *selector*, which implements step 3.

3.1 The solver

We know of two algorithms that can solve formal analogical equations on strings. The algorithm proposed by Lepage (1998) consists in computing two edit distance tables, one between the first and the second strings and one between the first and the third strings, then to synchronize these two tables thanks to an algorithm compactly described by the author. This is the algorithm used by Langlais and Patry (2007) in their study on translating unknown-words.

An alternative algorithm is due to Yvon (2003). It involves two operations on languages, namely the *shuffle* and the *complement* that can both be implemented by a finite-state automaton.

The shuffle of two strings *w* and *v* (*w* ∘ *v*) is the regular language gathering the strings obtained by selecting alternatively in *w* and *v* (without replacement) sequences of characters in a left-to-right manner (*e.g.*, $\underline{spondy}ondonti\underline{lalgia}tis$ and $ond\underline{spond}on\underline{ylal}titis\underline{gia}$ are two strings belonging to $\underline{spondylalgia} \circ ondontitis$). The complementary set of *w* with respect to *v* (denoted *w* \ *v*) is the set of strings formed by removing from *w*, in a left-to-right fashion, the symbols in *v* (*e.g.* $spondylitis$ and $spydoniltis$ are belonging to $spondyondontilalgiatis \setminus ondontalgia$).

Stroppa and Yvon (2005) sketched how a transducer can be built to recognize all the solutions to an analogical equation $[x : y = z : ?]$, that is, those belonging to {y ∘ z}\ x, where the two operations are naturally extended to handle regular sets. Our solver is described in Algorithm 1 and can be thought of as

[1] Those forms are French/English medical (single) terms.

s=10	4/7	(sponidylte,4) (itspndyloe,2) (itspondyle,2) (spondyilte,2) (spoindtyle,1)
s=20	8/14	(spondylite,4) (sponidylte,4) (spndyloite,4) (sponitedyl,3) (itspndyloe,2)
s=100	23/47	(spondylite,24) (itspndyloe,7) (itspondyle,7) (spndyloite,6) (sponitdyle,6)
s=1000	144/239	(spondylite,156) (ispondylte,41) (ispndylote,39) (spndyloite,39) (itespondyl,38)

Figure 1: The 5-most frequent solutions generated by our solver, for different sampling rates, for the equation [*chondropathie* : *spondylopathie* = *chondrite* : ?] with the frequency with which they have been generated (different samplings may lead to the same solution). n/t in the second column stands for the number n of forms that are generated more than once over the total number t of solutions generated.

a straightforward way of simulating this automaton.

Since the cardinality of the shuffle of two strings y and z grows exponentially in the length of those strings, we control the time response of our solver by sampling s strings from the language $y \circ z$.

An excerpt of the output produced by our solver for the equation [*chondropathie* : *spondylopathie* = *chondrite* : ?] is reported in Figure 1. We observe that by increasing the sampling rate s, the solver generates more (mostly spurious) solutions, but also increases the frequency with which the expected one is generated.

```
Input: $\langle x, y, z \rangle$, a triplet, s the sampling rate
Output: $sol$ a set of solutions to $[x : y = z : ?]$
  $sol \leftarrow \phi$
  for $i \leftarrow 1$ to s do
    $m \leftarrow$ shuffle($y$,$z$)
    $c \leftarrow$ complementary($m$,$x$)
    $sol \leftarrow sol \cup c$
  return $sol$
```

Algorithm 1: A Stroppa&Yvon flavored solver. `shuffle(y,z)` is randomly picking one element in $y \circ z$. `complementary(m,x)` returns the set of forms belonging to $m \setminus x$.

3.2 The generator

Identifying the stems of an unknown (source) term t during step 1 potentially requires to examine all possible triples between known terms in the input space: a naive implementation therefore has a cubic complexity in the cardinality of the training set $\mathcal{I}$. We applied the strategy proposed by Langlais and Patry (2007) for reducing this to a quadratic search procedure.

This strategy consists in repeatedly sampling pairs $\langle x, y \rangle$ in $\mathcal{I}$ and solving $[y : x = t : ?]$. Those solutions z that belong to $\mathcal{I}$ are defining the triplets $\langle x, y, z \rangle$ that will be considered during step 2. To further reduce the search, x are selected from the n-closest forms to t, and similarly, y are considered among the n-closest forms to x. Here, closeness is defined according to the conventional edit-distance.

With this simple strategy, the generator solves a number of source equations that is quadratic in n (chosen to be much smaller than the size of the source vocabulary $|\mathcal{L}|$). Since in our corpus (see section 4.1) most of the term only have one translation, the same number of equations has to be solved on the target side.

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. Various solutions have been proposed for this step, most of them relying on heuristically defined criterion, such as choosing the candidate of lesser degree, or the candidate which is supported by the largest number of triplets. The solution we propose here is novel and implies the use a classifier that will learn to select the good candidates.

3.3.1 The classifier

To this end, we trained, in a supervised manner, a binary classifier aimed at sorting out good translation candidates (as defined by a reference) from spurious ones. We applied the *voted-perceptron* algorithm originally introduced in 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) and are very simple to train.

In a nutshell, a weighted pool of perceptrons $\{(\mathbf{v}_k, c_k)\}_k$ is incrementally acquired during a batch training procedure sketched in Algorithm 2. Each perceptron $\mathbf{v}_k$ is parameterized by a vector in R^n (one component per feature on which we train the classifier), and is given a weight c_k, computed as the number of successive training examples it could correctly classifies before making a prediction error. c_k is thus a gross measure of the "goodness" of the k^{th} classifier. When the current perceptron misclassifies a training example, a new one is added to the pool; the coefficients of this new classifier are derived from the current perceptron according to a simple delta-rule and are kept fixed for the rest of the training procedure.

```
Input: $\{(\mathbf{x}_i, y_i)\}_{i \in [1,L]}$ where $y_i \in \{+1, -1\}$
Output: a pool of $K$ perceptrons $\{(\mathbf{v}_k, c_k)\}_{k \in [1,K]}$
  $k, c_1 \leftarrow 0$
  $\mathbf{v}_1 \leftarrow [0...0]^T$
  for all epoch do
    for all $i \in [1, L]$ do
      $\hat{y} \leftarrow sign(\mathbf{v}_k.\mathbf{x}_i)$
      if $\hat{y} \neq y_i$ then
        $\mathbf{v}_{k+1} \leftarrow \mathbf{v}_k + y_i \mathbf{x}_i$
        $c_{k+1} \leftarrow 1$
        $k \leftarrow k + 1$
      else
        $c_k \leftarrow c_k + 1$
```

Algorithm 2: Training regimen of the voted-perceptron algorithm.

At test time, a given observation $\mathbf{x} \in R^n$ is classified by averaging the prediction of the perceptrons in the pool, where the contribution of each perceptron $\mathbf{v}_k$ is weighted by c_k:

$$\hat{y} = sign\left(\sum_k c_k.sign(\mathbf{v}_k.\mathbf{x})\right)$$

3.3.2 The feature set

Training such a classifier is mainly a matter of feature engineering. In what follows, what we call an *example* is a pair of source-target analogical relations $(r, \hat{r})$ identified by the generator which proposes $\dot{t}$ as a candidate translation for the source term t:

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

.

Some features we consider depend on some structures compiled before feature extraction takes place. In particular, two codebooks C_s and C_t gather the most frequent cofactors involved in respectively the source and target analogies of the examples identified by the generator over the development material.

The most frequent entries of both codebooks are reported in Figure 2. Some cofactors, *e.g.* $\langle \epsilon, \textit{anti} \rangle$, capture the fact that complex terms in medical domain can be formed by derivation, that is, by adding an affix to a base-word, as in *antityphoid*. Some others such as $\langle \textit{ectomie}, \textit{otomie} \rangle$ reveal alternations that take place while compounding two (or more) components such as in *prostatectomie* or *prostatotomie*. Note that although the forms captured by frequent cofactors often correspond to morphemes such as the one used in (Déjean et al., 2002) or to the combining forms used in (Deléger et al., 2007), this is not necessarily the case.

source	$\langle \epsilon, \textit{péri} \rangle$ $\langle \epsilon, \textit{a} \rangle$ $\langle \textit{ectomie}, \textit{ite} \rangle$ $\langle \textit{ectomie}, \textit{otomie} \rangle$ $\langle \epsilon, \textit{anti} \rangle$
target	$\langle \epsilon, \textit{peri} \rangle$ $\langle \epsilon, \textit{a} \rangle$ $\langle \textit{ectomy}, \textit{itis} \rangle$ $\langle \textit{ectomy}, \textit{otomy} \rangle$ $\langle \epsilon, \textit{anti} \rangle$

Figure 2: The 5 most-frequent cofactors of the source and target codebooks.

We split the range of the frequency of the cofactors in C_s and C_t into two uniform sets of f bins b_s and b_t. We also trained a target character-based m-gram model L_m on the terms of the training material (see SEARCH in Section 4.1).

We considered the following feature sets for characterizing each example e:

[deg] Degrees of the source and target analogies of e.

[freq] The frequency $f_{\dot{t}}$ with which a candidate $\dot{t}$ has been generated for the translation of a given term t, as well as the *rank* of $\dot{t}$ in the list of candidates sorted in decreasing order of frequency.

[book-b] A source vector $c_s = [c^s_1, \ldots, c^s_b]$ where c^s_i is set to 1 if the i^{th} cofactor of codebook C_s is present in the list of cofactors of e, 0 otherwise. A similar vector c_t is computed on the target side.

[lm-m] The minimum and average probabilities computed by L_m for $\dot{t}$, and the difference of both.

[bin-f] A source vector of dimension f which keeps track of the count of the number of source cofactors in e which frequency falls into the i^{th} bin of b_s. A similar vector is computed on the target side.

4 Experimental protocol

4.1 The corpus

In this work, we concentrated on translating simple terms belonging to the biomedical domain from French into English. We used a list of French terms and their authoritative English translations extracted from the online medical dictionary *Masson*.[2] The same list was used in the work of Claveau and Zweigenbaum (2005).

We only focused on the pairs of terms which normalized edit-distance were ranging from 0.02 to 0.67, corresponding respectively to pairs that differ by only one character, such as $(\textit{artériorrhexis}, \textit{arteriorrhexis})$, to some rather distant pairs, such as $(\textit{toux}, \textit{cough})$ or $(\textit{grossesse}, \textit{pregnancy})$. A few terms are containing uppercase letters, we did not lowercase them.

We randomly split the 13 392 pairs of terms into SEARCH (80%), DEV (10%), and TEST (10%) sets, used respectively to train the generator, to train the selector(s), and to test the complete system. We further split the material used to train the selector(s) into DEV-TRAIN (90%) and DEV-TEST (10%).

4.2 Evaluation

Let $\mathcal{S} \equiv \{S_i : i \in [1, N]\}$ be the set of N (source) terms we seek to translate. For each term S_i, the generator produces a set of N_i candidate translations $T_i \equiv \{T_{ij} : j \in [0, N_i]\}$. The selector can be seen as a binary classifier[3] producing a decision $b_{ij} \in \{0, 1\}$ for each candidate T_{ij}, on the face of which translations $T_i^+ \equiv \{T_{ij} : b_{ij} = 1\}$ will be proposed; that is, those for which b_{ij} equals 1.

Let $\mathcal{R} \equiv \{r_i \equiv \{r_{ik}\} : k \in [1, n_i]\}$ be the reference set, that is, the set of n_i reference translations for each source term S_i. In this study, n_i usually equals 1, but it happens occasionally that a term receives several reference translations.

We measure the overall performance of `AnaTerm` with micro and macro f-measures. The micro-values are computed for each unknown term, and then averaged:

$$\begin{array}{lcl} \cap_i & = & r_i \cap T_i^+ \\ \text{m-precision} & = & \frac{1}{N}\left(\sum_{i=1}^{N} \mid \cap_i \mid / |T_i^+|\right) \\ \text{m-recall} & = & \frac{1}{N}\left(\sum_{i=1}^{N} \mid \cap_i \mid / |r_i|\right) \end{array}$$

Macro-values are computed by keeping the count of good translations produced over a session, and by normalizing adequately:

$$\begin{array}{lcl@{\qquad}lcl} I & = & \sum_{i=1}^{N} \mid \cap_i \mid & \text{M-precision} & = & I/P \\ P & = & \sum_{i=1}^{N} T_i^+ & \text{M-recall} & = & I/R \\ R & = & \sum_{i=1}^{N} n_i & & & \end{array}$$

We are also interested in evaluating the selector component alone. For that purpose, we measure the accuracy of a selector as the ratio of good decisions (positive or not) made by the classifier:

$$acc = \left[\sum_{i=1}^{N}\sum_{j=1}^{N_i} \delta\left(b_{ij} = \delta\left(T_{ij} \in r_i\right)\right)\right] / \sum_{i=1}^{N} N_i$$

$\delta(\bullet)$ is 1 when $\bullet$ is true, and 0 otherwise. At a finer grain, we also measure the accuracy on good ($acc+$) and bad examples separately.

5 Experiments

5.1 The generator

We first investigate the impact of the few parameters controlling the generator; namely, the sampling rate `s`, and the number `n` of neighbors considered. Performances are measured on TEST with an *oracle* selector which simply looks at the reference in order to decide which candidate translation is positive. Micro and macro f-measures of variants where `s` ranges from `50` to `2000`, and `n` from 20 to 300, are reported in Figure 3. We observe that the main factor impacting the generator is `n`, the number of neighbors considered during Step 1: the more the neighbors, the better the performance. For the variant `s=250` for instance, the macro f-measure goes from 62.9% for `n=20` to 78.4% for `n=300`. The influence of the sampling parameter is rather small.

[2] Available at url `http://www.atmedia.com`.

[3] Note that this does not impose that the selector is being implemented as a binary classifier.

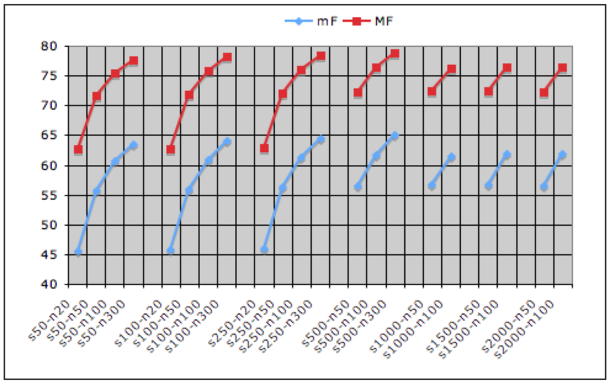

Figure 3: Oracle performance of several configurations measured on the TEST material (1306 terms).

The best variant tested (s=500-n=300) records a micro-fmeasure of 65% and a macro-fmeasure of 78.8%. Since with an oracle selector, the macro-precision equals 100, it means that only 65% (849 out of 1306) of the terms received a translation. 250 (19%) of the 1306 source terms of the test material were not translated because of a failure during Step 1; another 35 because of a failure during Step 2. Among the terms that did not receive by AnaTerm a valid translation, 62.4% did not receive any candidate at all, the others (37.6%) receiving an average of 6832 candidates (231 259 at most). Increasing n will likely reduce the number of terms without any match during the first step, at the expense of time.

From Figure 4, we see that the ratio of terms that receive at least one candidate translation by the generator decreases with the (normalized) edit-distance between a source term and its reference translation.

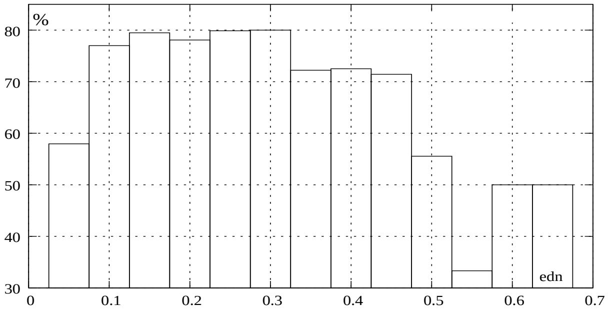

Figure 4: Ratio of the terms of the DEV material (1271 terms) that receive a solution by the generator, as a function of the normalized edit-distance between the source term and its translation.

In the remainder, we concentrate on the configuration s=100-n=300, which shows good performance while generating a (barely) decent number of examples. In fact, from the 1 271 terms of DEV, this configuration generated over 4 millions of examples (more than 3.6 serving as training material for the selectors); and over 4.6 millions of examples from the 1306 terms of the TEST set. Less than 1% of those examples are positive ones.

5.2 The selector

We trained different classifiers on the 3.6 millions of examples of the DEV-TRAIN set. Each classifier was trained using 40 epochs.[4] The performance of each variant on both DEV-TEST and TEST are reported in Table 1. Note that for those feature sets which depend on a parameter, such as lm, bin, or book, we report only the best variant we observed (as defined by *acc* measured on the training set). For instance, using a trigram language model leads to better performance than a 4-gram model, which, in turn, outperforms a bigram model.

For comparison purposes, we implemented two baselines: all-spurious which classifies as spurious all examples (note that this would result in a totally silent overall system), and top-freq which classifies as good, the only examples leading to a candidate which frequency is maximum. Most of the classifiers we trained clearly outperform them.

Two feature sets outperform the others: deg and bin. The language model feature-set alone is poor but happens to lead, in conjunction to deg, to one of our best classifier. Currently, the best variant correctly classifies 99.9% of the examples of TEST; 89.6% of the positive ones being correctly identified (recall that there are less than 1% of positive examples in our test sets).

The fact that the classifiers perform much better on TEST than on DEV-TEST suggests that we should backup our observations by cross-validation. Other combinations of features will likely yield better classification rates; but classifying roughly 90% of the positive examples as such is already what we consider a good performance, suitable for the investiga-

[4] We analyzed the learning curves of some classifiers trained on a balanced corpus (that is, with as many good and spurious examples), and noticed that, depending on the feature set, some over-training seems to occur after several tens of iterations.

	DEV-TEST		TEST	
feature sets	acc	acc+	acc	acc+
book200 (b200)	98.65	9.87	98.79	16.44
freq	98.81	38.31	99.47	63.47
lm3 (l3)	99.14	40.40	98.97	39.35
bin3 (f3)	99.49	60.84	99.82	85.82
deg (d)	99.49	60.90	99.82	85.75
d-l3-b200	99.54	65.04	99.78	85.98
d-l3-f5	99.53	65.56	99.77	85.81
d-l3-f100-b100	99.55	65.57	99.78	86.45
d-l3	99.53	65.89	99.76	85.90
d-l3-freq-f50	99.60	67.45	99.86	89.64
all-spurious	99.23	0	99.14	0
top-freq	96.59	17.89	99.14	44.67

Table 1: Performance of several classifiers trained on DEV-TRAIN and tested against DEV-TEST and TEST. The top figures concern single sets of features; the bottom one, the 5-best feature-set combinations we computed so far.

tions carried out in the next section.

5.3 The overall system

We now evaluate AnaTerm as a translation device. In order to appreciate the contribution of the selectors to the overall system, we considered two oracle ones; oracle which simply looks at the reference in order to decide which candidate translation is positive; and voc which assumes that the target vocabulary is known, and which selects the only candidates that belong to this vocabulary.[5]

We also implemented three baselines by selecting the candidate form(s) with maximum frequency (top$_{freq}$) or 3-gram log-probability (top$_{lm}$) or minimum edit-distance to the source term (top$_{ed}$); the latter being motivated by the fact that in the medical domain, a French term and its English translation are often close. The performance of the five-best selectors we trained, plus those of the baselines are reported in Table 2.

Each decision taken by the selector is independent of the other decisions. It is therefore not surprising that some terms receive more than one candidate translation, which decreases (micro and macro) precision. We can further filter out some candidates by the means of other criteria. For instance, we can select the top-frequent solution(s) ($_{freq}$), the best ranked one(s) according to the language model ($_{lm}$), or the closest to the source term as defined by edit-distance ($_{ed}$).

By this mean, we increase precision without significantly impacting recall. Roughly 64% of the terms belonging to TEST receive at least one candidate translation by d-l3-freq-f50$_{freq}$ (1.07 on average), and in 80% of the cases, the reference translation is among the list of (at most 9) candidates. A random excerpt of candidate terms generated by the variant d-l3-freq-f50 are reported in Figure 5.

A more natural way to overcome this cascade of selectors would be to directly train a *reranker* to do the job. We could apply for that the reranking strategy described in (Collins and Duffy, 2002) which introduces only minor modifications to the training regimen we considered in this study. This is left as future work.

5.4 A point of comparison

To put these figures in perspective, we used the bitext from the medical domain that Langlais et al. (2006) used for adapting a statistical translation engine to the medical domain. This bitext gathers over 800 000 pairs of sentences collected from 20 000 pages downloaded from the website of Health Canada[6] and 14 000 pages from the website of the public Health Agency of Canada.[7].

Only 311 (23.8 %) of the source terms of TEST are present in this resource which gathers roughly 150 000 word-forms per language. For 68 of these terms, the reference translation is not present in the target material, which means that at best we could identify the translation of 18.6% of our test corpus.

6 Related Work

Chiao and Zweigenbaum (2002) describe an experiment where a comparable corpus of the medical domain of above 600 000 words in French and En-

[5] This last variant may be of practical interest, when we do have a target vocabulary from which we would like to identify translations. This was for instance the scenario investigated in (Langlais and Patry, 2007) where the authors computed a target vocabulary from a large repository of target language texts.

[6] http://www.hc-sc.gc.ca

[7] http://www.phac-aspc.gc.ca

config	ans.	hits	mP	mR	mF	MP	MR	MF	avr	max
oracle	837	837	64.09	64.09	64.09	100.00	64.09	78.11	1	1
voc	845	837	60.25	64.09	62.11	87.83	64.09	74.10	1.13	3
top$_{lm}$	1003	185	14.13	14.17	14.15	18.39	14.17	16.00	1.00	2
top$_{ed}$	1003	610	30.23	46.71	36.70	18.73	46.71	26.74	3.25	52
top$_{freq}$	1003	755	33.36	57.81	42.30	4.68	57.81	8.66	16.09	999
d-l3-b200	733	615	32.33	47.09	38.34	41.16	47.09	43.93	2.04	24
d-l3-f5	724	599	31.83	45.87	37.58	42.04	45.87	43.87	1.97	21
d-l3-f100-b100	760	642	34.71	49.16	40.69	44.25	49.16	46.57	1.91	21
d-l3	718	589	31.32	45.10	36.97	41.42	45.10	43.18	1.98	23
d-l3-freq-f50	835	677	47.60	51.84	49.63	65.66	51.84	57.94	1.23	10
d-l3-freq-f50$_{lm}$	835	642	49.16	49.16	49.16	76.89	49.16	59.97	1.00	1
d-l3-freq-f50$_{ed}$	835	668	50.80	51.15	50.98	77.76	51.15	61.71	1.03	5
d-l3-freq-f50$_{freq}$	835	669	50.23	51.23	50.72	74.58	51.23	60.74	1.07	9

Table 2: Performance measured on the TEST material for several variants of AnaTerm sharing the generator s=100-n=300. The top part of the Table are oracle and baseline variants; the bottom part are the different variants we tried. $ans.$ (resp. $hits$) stands for the number of source terms with at least one (resp. correct) candidate translation; avr (resp. max) stands for the average (resp. maximum) number of candidates received per source term (when at least one is found).

chondromyxosarcome	(***chondromyxosarcoma***,59) (myxochondrosarcoma,51) (myxosarcochondroma,41)
électroradiologie	(***electroradiology***,26) (radioelectrology,24)
pathogène	(**pathogenic**,43) (*pathogenous*,34) (pathogen,31)
périlobulite	(lobulitiperis,65) (***perilobulitis***,65) (lobulitisperi,65) (lobuliperitis,64)

Figure 5: Solutions proposed by d-l3-freq-f50, with their frequency, for some source terms. The reference solution is in bold; the candidates in italic are those selected by the variant d-l3-freq-f50$_{lm}$.

glish has been collected as well as a seed bilingual lexicon of more than 18 000 entries. They report that a context-vector approach allowed to identify in the top 10 candidates the sanctioned translation of an unknown term, 50% of the time. With the goal of illustrating the comparability of non-parallel corpora, Morin et al. (2007) report that a similar approach could translate 51% of single terms and 49% multi-word terms when a list of top 10 candidates is considered, making use of a comparable corpus on the domain of nutrition (above 700 000 French and 807 000 Japanese words), and a bilingual seed bilingual lexicon of above 173 000 entries.

Claveau and Zweigenbaum (2005) trained a transducer from a set of pairs of terms. They report precision rates ranging from 52% to 67%, when trained on a set of 3 000 training pairs, translating from French into English, and vice-versa. Claveau (2007) proposed to learn rewrite rules from the character-based alignment of the pairs of terms in the training material. At translation time, all the rules which source part matches the source term are tested and the resulting candidate translations are ranked by a language model. They report an improvement of roughly 10% over the previous approach if 11 000 pairs of terms are used for training.

This last two approaches are very close in spirit to the one we propose since they are exploiting no other resource than a specialized bilingual seed lexicon. Among the differences, however, we must stress that AnaTerm does not rely on any alignment step. Instead, the translations emerge from a general principle: proportional analogy. Due to different experimental settings, however, we can not compare them precisely. In particular, we did not consider in our experiments terms that were translated verbatim, and we did not focus on the only terms than contain at least 8 characters, as was done in the other two studies. This is part of future work to carry out a more careful comparison of our approach with the one of (Claveau, 2007).

7 Discussion and future work

We investigated the use of analogical learning (Stroppa and Yvon, 2005) to the task of translating from French into English single terms of the medical domain. We show that it is possible to train a classifier to distinguish good analogies from spurious ones, an improvement over the approach of Langlais and Patry (2007).

We analyzed the impact of different factors on the performance of `AnaTerm`, and observed that one of the best variant we tested so far could propose a candidate translation for 64% of the terms, with a correct translation in 80% of the cases.

We already mentioned several enhancements to the approach we presented in this study, that we are currently investigating. There are other avenues we plan to investigate. First, since we observed that widening the search space leads to improvements, we must investigate ways to speed-up the search procedure. We can certainly improve the implementation of the solver used in this study, but the main challenge remains to identify fruitful input triplets in the first place. Second, we would like to investigate how `AnaTerm` scales to multi-terms and to different domains.

Acknowledgements

Part of this work has been accomplished while the first author was visiting the department of Computer Science and Networks at Telecom ParisTech, Paris.

References

David A. Aha. 1997. Editorial. *Artificial Intelligence Review*, 11(1-5):7–10. Special Issue on Lazy Learning.

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.

Y-C. Chiao and P. Zweigenbaum. 2002. Looking for candidate translational equivalents in specialized, comparable corpora. In *19th COLING*, Taipei, Taiwan.

V. Claveau and P. Zweigenbaum. 2005. Automatic translation of biomedical terms by supervised transducer inference. In *10th AIME*, Aberdeen, Scotland.

V. Claveau. 2007. Traduction automatique de termes biomédicaux pour la recherche d'information interlingue. In *4th CORIA*, Saint-Étienne, France.

M. Collins and N. Duffy. 2002. New ranking algorithms for parsing and tagging: Kernels over discrete structures. In *40th Annual Meeting of the ACL*, pages 263–270, Philadelphia, PA.

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

H. Déjean, É. Gaussier, and F. Sadat. 2002. An approach based on multilingual thesauri and model combination for bilingual lexicon extraction. In *19th COLING*, Taipei, Taiwan.

L. Deléger, F. Namer, and P. Zweigenbaum. 2007. Defining medical words: Transposing morphosemantic analysis from french to english. In *MEDINFO*, pages 152–156, Brisbane, Australia.

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. Langlais and M. Carl. 2004. General-purpose statistical translation engine and domain specific texts: would it work? *Terminology*, 10(1):131–152.

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

P. Langlais, F. Gotti, and A. Patry. 2006. De la chambre des communes à la chambre d'isolement: adaptabilité d'un système de traduction basé sur les segments. In *Proceedings of 13th TALN*, pages 217–226, Leuven, Belgium.

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.

F. Yvon. 2003. Finite-state transducers solving analogies on words. Technical Report 2003D008, ENST.

Dépôt légal : 2008 – 4ème trimestre
Imprimé à l'Ecole Nationale Supérieure des Télécommunications – Paris
ISSN 0751-1345 ENST D (Paris) (France 1983-9999)

TELECOM ParisTech

Institut TELECOM - membre de ParisTech

46, rue Barrault - 75634 Paris Cedex 13 - Tél. + 33 (0)1 45 81 77 77 - www.telecom-paristech.frfr

Département INFRES