Saussurian analogy: a theoretical account and its application

Yves Lepage & Ando Shin-ichi
ATR Interpreting Telecommunications Research Labs,
Hikaridai 2-2, Seika-cho, Soraku-gun, Kyoto 619-02, Japan
{lepage,ando}@itl.atr.co.jp

Abstract

In the *Cours de linguistique générale*, Saussure mentions a phenomenon of tremendous importance in language, *analogy*. For example, given the series *walk*, *walked* and *look*, how can we coin the fourth term, *looked*? We give a possible account of this phenomenon in terms of edition distances, thus paving the way to computational applications. This explanation accounts for prefixing, suffixing and infixing. We show how it is possible to perform the analogical analysis and generation of sentences, using a tree-bank and approximate pattern-matching. As a consequence, our proposal finds its place in the example-based approach to natural language processing.

1 Introduction

In the *Cours de linguistique général*, which dates back to 1916, Saussure mentions a phenomenon of tremendous importance in language, *analogy*: given some series of three words, human beings are able to coin a fourth one. One can see a reactualisation of this principle in the example-based approach to machine translation. Analogy seems to have never been theorised in a monolingual framework, making its bilingual application questionable. The purpose of this article is to propose a possible, mathematically sound explanation, and to show the path to computational applications.

2 Saussurian analogy

In Chapter four, Part III of the *Cours de linguistique générale*[1], Saussure points out what he calls *analogy*: given two forms of a given word, and only one form of a second word, it is possible to coin the missing form[2].

Latin: *ōratōrem* : *ōrātor* = *honōrem* : **x**
x = *honor*

In this particular case, Saussure was interested in explaining the competition of *honor* with the older form *honos*. *honor* is not a phonetic transformation of *honos* by rhotacism, but simply the result of analogy.

Analogy is very general, and its effects are seen in a number of other places. It may explain all flexional paradigms, from conjugation to declension[3].

German: *setzen* : *setzte* = *lachen* : **x**
x = *lachte*

Analogy also explains what is called the productivity of language, *i.e.*, the fact that understandable words can be coined, which are not registered in dictionaries, and may have never been uttered before by the speaker nor heard before by the listener[4].

French: *réaction* : *réactionnaire* = *répression* : **x**
x = *répressionnaire*

Finally, analogy also explains incorrect forms or *barbarisms*, examples of which are frequent in child language[5].

French: *éteindrai* : *éteindre* = *viendrai* : **x**
x = *viendre*

Our goal is to give one possible account of this phenomenon in computational terms, and to show that, given a tree-bank, a possible application may be the analysis or generation of sentences.

[1] All examples in this section are from the *Cours*.

[2] *ōrātor* (orator, speaker) and *honor* (honour) nominative singular, *ōratorem* and *honōrem* accusative singular.

[3] *lachen* (to laugh) and *setzen* (to put), *lachte*, *setzte* past forms.

[4] *réaction* (reaction) and *répression* (repression) nouns, *réactionnaire* (reactionary) adjective; *répressionnaire* sounds perfectly French, but will not be found in a dictionary.

[5] *éteindre* (to extinguish; to turn off) infinitive, *éteindrai* and *viendrai* future tense; *viendre* is a barbarism in place of *venir* (to come) (compare, in English, *goed* for *went*).

3 A possible account

3.1 Notations

Let $\mathcal{V}$ be a non-empty finite set, called the vocabulary. $(\mathcal{V}^*, .)$ is the monoid over $\mathcal{V}$ where . denotes concatenation. $\mathcal{V}^*$ is also the infinite union of all $\mathcal{V}^n$ for $n \in \mathbb{N}$. By convention, $\mathcal{V}^0 = \{\varepsilon\}$ with ε being the empty string.

Using these notations, analogies are equations with one unknown on $\mathcal{V}^*$: $u : v = w : \mathtt{x}$. To be able to solve analogies, it is necessary to give a meaning to such a notation.

3.2 A geometrical view

In our attempt to discover a mathematical explanation of analogy, we were long hindered by the notation itself. Of course, the idea behind it is that analogy could be considered a similar psychological process as the one intervening in proportions:

$$\begin{array}{ccc} \textit{mathematics} : \textit{mathematical} & = & \textit{physics} : \mathtt{x} \\ 2 : 4 & = & 3 : \mathtt{x} \end{array}$$

But $\mathbb{Q}$, the set of rationals, is mathematically well equipped. Addition defines a commutative group, and multiplication makes it a field. Proportions in $\mathbb{Q}$ are thus well understood, and safely solved. What is true for $\mathbb{Q}$ is not for $\mathcal{V}^*$. The basic operation, concatenation, is not commutative and does not define a group, but a relaxed structure, that of a monoid. And no one knows what the meaning of $u : v$ could possibly be.

In fact, looking at analogy from the previous point of view is misleading because, intentionally or not, we think of numbers, which enforces too many constraints. A better, more relaxed view of the problem is that of a rectangle. In a rectangle, opposite sides and diagonals are equal (see Figure 1).

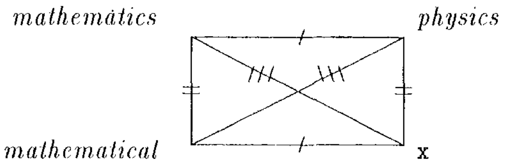


Figure 1: Analogy seen as a rectangle

3.3 Formalisation

This view makes explicit that analogy sets a relation between an unknown on one hand, and three terms on the other hand. Now, carrying on with the geometrical parallel, analogy may be interpreted in terms of distances as follows : the distance of any term to the unknown is the same as the distance between the two remaining terms. We thus posit the following equivalence.

Definition 1 (Analogy)

$$u : v = w : \mathtt{x} \overset{\Delta}{\Longleftrightarrow} \left\{ \begin{array}{rcl} \mathrm{dist}(u, v) & = & \mathrm{dist}(w, \mathtt{x}) \\ \mathrm{dist}(u, w) & = & \mathrm{dist}(v, \mathtt{x}) \\ \mathrm{dist}(v, w) & = & \mathrm{dist}(u, \mathtt{x}) \end{array} \right.$$

The rectangle view does not forbid commutativity for *dist*, a notable difference with division on numbers, where 2/4 is not the same as 4/2.

3.3.1 Linguistic interpretation

Let us linguistically interpret the previous system of equations. Suppose we get the following analogy to solve: *mathematics* : *mathematical* = *physics* : x. Of course, x = *physical*.

The first two equations show that the terms on the diagonal may be exchanged. A linguistic interpretation is that analogy involves two orthogonal dimensions reflecting the duality of the lexeme/morpheme (or root/affix, or meaning/function, *etc.*) separation.

$$\begin{array}{l} \mathrm{dist}(\textit{mathematics}, \textit{mathematical}) = \\ \qquad\qquad \mathrm{dist}(\textit{physics}, \textit{physical}) \\ \mathrm{dist}(\textit{mathematics}, \textit{physics}) = \\ \qquad\qquad \mathrm{dist}(\textit{mathematical}, \textit{physical}) \end{array}$$

On each side of the equal sign something is conserved (one dimension), and something changes (second dimension).

- In the example, the first equation stands for a conservation in meaning ("**mathematics**" as opposed to "**physics**") and a change in categories,
- whereas the second equation stands for a conservation of grammatical categories (N as opposed to A), but a change in meanings.

The third equation means that, somehow, analogy neutralises changes performed at the same time along the two previous dimensions.

$$\begin{array}{l} \mathrm{dist}(\textit{mathematics}, \textit{physical}) = \\ \qquad\qquad \mathrm{dist}(\textit{physics}, \textit{mathematical}) \end{array}$$

On each side of the equality sign, both changes in meanings and categories, performed at the same time, leave the proportion unchanged.

3.3.2 Complete formalisation

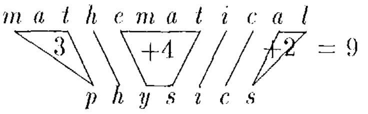
In order to complete the formalisation, *dist* remains to be defined. Edition distances which have been proposed in many works (Levenshtein 65), (Wagner & Fischer 74), (Selkow 77), *etc.*, are a good candidate. They are mathematically sound as well as intuitively relevant: they reflect a sensible notion, that of keystrokes, and turn out to be metrics under some hypotheses. They answer the correction problem: *what is the minimal number of edit operations needed to transform one word into another one?* In our example, how many characters need to be changed to transform *mathematical* into *physics*? Edit operations are insertion (for instance, $\varepsilon \rightarrow p$), deletion (like $l \rightarrow \varepsilon$)

and replacement (like $a \rightarrow s$). A distance can be defined by assigning weights to these three operations, 1 for each of them, for simplification. The edit distance is then a simple extension from edit operations to strings.

Definition 2 (Edition distance) *Let $\mathcal{V}$ be a vocabulary,* dist *is defined on $\mathcal{V}^*$ as a commutative operation, in the following way:*
$\forall (a,b) \in \mathcal{V}^2$, $\forall (u,v) \in (\mathcal{V}^*)^2$,

$$\begin{array}{rcl}
\mathrm{dist}(\varepsilon,\varepsilon) & = & \mathrm{dist}(a,a) = 0 \\
\mathrm{dist}(\varepsilon,a) & = & \mathrm{dist}(a,b) = 1 \ \textit{if}\ a \neq b \\
\mathrm{dist}(a.u,\varepsilon) & = & \mathrm{dist}(a,\varepsilon) + \mathrm{dist}(u,\varepsilon) \\
\mathrm{dist}(a.u,b.v) & = & min(\begin{array}{l} \mathrm{dist}(a,\varepsilon) + \mathrm{dist}(u,b.v), \\ \mathrm{dist}(a,b) + \mathrm{dist}(u,v), \\ \mathrm{dist}(\varepsilon,b) + \mathrm{dist}(a.u,v) \end{array})
\end{array}$$

With this definition and a weight of 1 for each of the three edit operations, the distance between *mathematical* and *physics* becomes 9.



As a mathematical result, with more general weights, it can be proved that, if the edit operations define a metric on $\mathcal{V} \cup \{\varepsilon\}$, then the edit distance on $\mathcal{V}^*$ is also a metric. We recall the formal definition of a metric.

Definition 3 (Metric) *Let $\mathcal{S}$ be a set,* dist *a function from $\mathcal{S} \times \mathcal{S}$ to $\mathbb{R}^+$, the set of non-negative real numbers,* dist *is a metric on $\mathcal{S}$ if and only if*

- *(equality)*
  $\forall (a,b) \in \mathcal{S}^2$, $\mathrm{dist}(a,b) = 0 \Leftrightarrow a = b$
- *(commutativity)*
  $\forall (a,b) \in \mathcal{S}^2$, $\mathrm{dist}(a,b) = \mathrm{dist}(b,a)$
- *(triangle inequality)* $\forall (a,b,c) \in \mathcal{S}^3$,
  $\mathrm{dist}(a,c) \leq \mathrm{dist}(a,b) + \mathrm{dist}(b,c)$

3.4 Coverage

Having defined what we understand by analogy in a formal way, we inspect some of its properties. We first make a very strong but necessary assumption about the nature of the solution of an analogy. Following the linguistic feeling, we impose that the solution of an analogy be built only with the elements of the vocabulary present in the three given terms. In other words, no material from outside should be used.

This constraint does not prevent analogies from having multiple solutions. It suffices that the distances become too large relative to the lengths of the words. $a : the = of : \mathbf{x}$ is such a case. The constraint eliminates, for instance, all words of the form *txy*, with x and y two letters outside of the set $\{a, e, f, h, o, t\}$, but does not bar *ttt*, *hhh*, *eee*, which are solutions of this analogy. But, as a matter of fact, this kind of example does not make much linguistic sense.

3.4.1 Equality

A degenerated case of analogy is when two of the three terms are equal. The only possible solution is then the third term. In other words, nothing new can really be said. This meets common sense.

$$\forall (u,v) \in (\mathcal{V}^*)^2,\ u : u = v : \mathbf{x} \ \Rightarrow\ \mathbf{x} = v$$

This property is always true. It is proved thanks to the equality property of a metric: $u : u = v : \mathbf{x}$ $\Rightarrow$ $\mathrm{dist}(u,u) = 0 = \mathrm{dist}(v,\mathbf{x})$ $\Rightarrow$ $\mathbf{x} = v$.

Some important linguistic phenomena are covered by our proposal for linguistic examples. But the corresponding mathematical properties appear not to hold in the general case. In fact, studying the necessary and sufficient conditions under which they are true remains an open problem. It seems that, in all cases, it has to do with some "weakest links" along the pair of strings considered (minimisation of a sum of distances).

3.4.2 Transitivity

An important property which works in many cases, and at least on linguistic examples, but may not be true in the general case[6], is transitivity:

$$u : v = u' : v' \ \wedge\ u' : v' = w : \mathbf{x} \ \Rightarrow\ u : v = w : \mathbf{x}$$

This accounts for the fact that any representative in a group of conjugation/declension/*etc.* may be chosen as the model. In Ancient Greek, *λογος* is always taken as a model for the declension of the 1st group of masculine nouns, although any other word from the same group would have been as good.

3.4.3 Prefixes and suffixes

Our definition of analogy fortunately captures linguistic cases where prefixes (or suffixes) are involved.

$$u.t : u.v = w.t : \mathbf{x} \ \Rightarrow\ \mathbf{x} = w.v$$

This is not true in the general case. At least, $\mathbf{x} = w.v$ always verifies the first two distance equations:

$$\left\{ \begin{array}{rcl}
\mathrm{dist}(u.t, u.v) & = & \mathrm{dist}(t,v) \\
 & = & \mathrm{dist}(w.t, w.v) \\
 & = & \mathrm{dist}(t,v) \\
 & & \\
\mathrm{dist}(u.t, w.t) & = & \mathrm{dist}(u,w) \\
 & = & \mathrm{dist}(u.v, w.v) \\
 & = & \mathrm{dist}(u,w)
\end{array} \right.$$

thanks to a property of edit distances, which we give here without a proof: $\forall\ (u,v,w)\ \in\ (\mathcal{V}^*)^3$,

[6] Counter-example: $the : ttt = a : of \ \wedge\ a : of = the : hhh \ \not\Rightarrow\ the : ttt = the : hhh$ because $\mathrm{dist}(the, the) = 0 \neq \mathrm{dist}(ttt, hhh) = 3$

$\mathrm{dist}(u.v, u.w) = \mathrm{dist}(v, w)$. But the third equation may not always be verified. A sufficient condition for it to hold is that the joints between prefixes and suffixes minimise some sums of distances:

$$\begin{aligned} \mathrm{dist}(u.v, w.t) &= \mathrm{dist}(u, w) + \mathrm{dist}(t, v) \\ &= \mathrm{dist}(u.t, w.v) \end{aligned}$$

This is the case when prefixes and suffixes are dissimilar enough, as in our example with *mathemat-i-cs* and *phys-i-cal*, but in the general case, only $\mathrm{dist}(u.v, w.t) \leq \mathrm{dist}(u, w) + \mathrm{dist}(v, t)$ holds.

3.4.4 Infixes and umlauts

Similarly to prefixing and suffixing, our formalisation accounts for linguistic examples of infixing, a phenomenon well illustrated by semitic languages[7] (here, the replacement of an *a* by an *i*).

Arabic: *arsala* : *mursilun* = *aslama* : **x**
x = *muslimun*

It also accounts for some (not all) examples of sound changes, like umlaut in German[8].

German: *Balg* : *Bälge* = *Hals* : **x**
x = *Hälse*

These linguistic cases work partly thanks to the previous property of distances with prefixes.

3.4.5 Reduplication

Unfortunately, our proposal does not render an account of *reduplication*. This would be necessary if we wanted to describe, for example, the formation of plurals in Malay/Indonesian: *orang* → *orang-orang*[9]. Here, a speculative remark would link the power of analogy with some class of languages; our proposal seems not to go beyond regular languages.

4 Application

In the sequel, we will apply the principle of analogy not on words anymore, but on sentences. In the same way as words are strings of characters, sentences are strings of words. So, the shift from words to sentences is just of matter of reformulation.

We also recall that edit distances and edit operations are not confined to strings; they extend in a natural way to forests, and hence to trees. In fact, it is possible to give a definition of an edit distance on forests which generalises the definitions on strings (Wagner & Fischer 74) and on trees (Selkow 77). Hence the possibility of applying analogy to trees.

The example-based approach in machine translation, inaugurated by (Nagao 84) and illustrated by (Sadler and Vendelmans 90) or (Sato 90), for instance, relies on the assumption that, if two sentences are "close", then, their analyses should be "close" too. By consequence, if the analysis of a first sentence is known, the analysis of the second one could be obtained by performing slight "modifications" on it. A problem arises: where are the slight "modifications" to be performed, and what are they? In that matter, edit distances could help a lot: "close" means at a distance not too large, and "modifications" are edit operations.

4.1 Analysis by analogy

4.1.1 Principle

Suppose we have a collection of sentences (a data-base) already analysed (in fact, a tree-bank). For a new sentence, called the *prototype*, our goal is to build its analysis, *i.e.*, a corresponding tree. Of course, the ideal case is when the prototype is already present in the tree-bank, which means that the analysis is found there too.

In general, the prototype will not be found in the tree-bank. The search may thus be relaxed to similar sentences. Now, if at least two different sentences are retrieved by approximate matching, a fourth one can be built by analogy. Figure 2 illustrates this: the prototype is in the upper left corner; the two sentences on its right and under it have been obtained by approximate matching. Knowing the respective distances between these three sentences (on the arrows), sentence **x** can be computed by analogy.

If by chance sentence **x** belongs to the tree-bank, its analysis is also in the tree-bank. Now, a reverse process on trees delivers an analysis for the prototype, as illustrated in Figure 3. The three trees in the right and bottom corners are the corresponding analyses of the sentences of Figure 2. They were taken from the tree-bank. The distances are given on the arrows. Tree **y** is the solution of the analogy, and we claim that it is the analysis of the prototype sentence.

4.1.2 Implementation

Approximate matching is retrieval of all sentences at a distance less than a threshold from a given prototype. Efficient algorithms, using dynamic programming, have been proposed to perform approximate matching (Ukkonen 83) and (Landau & Vishkin 88). Our method is somewhat different. We do as if we wanted to generate the entire set of sentences at a distance less than or equal to the threshold. In doing that, we introduce a *don't care* symbol representing any possible word. Pattern-matching with *don't care* symbols has already been studied (Pinter 85). Of

[7] *arsala* (he sent) and *aslama* (he became converted) verbs 3rd person singular past; *mursilun* (a sender) and *muslimun* (a convert) agent nouns.

[8] *Balg* (pelt, skin) and *Hals* (neck) singular; *Bälge* and *Hälse* plural.

[9] *orang* (human being) singular, *orang-orang* plural.

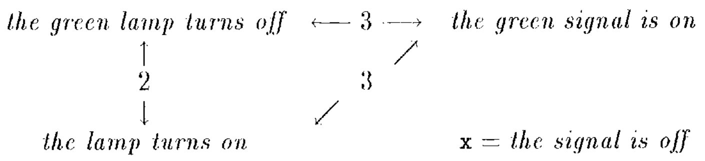


Figure 2: Prototype (upper left corner), sentences obtained by approximate matching and **x**, sentence obtained by analogy, and retrieved from the tree-bank. Distances are given in words.

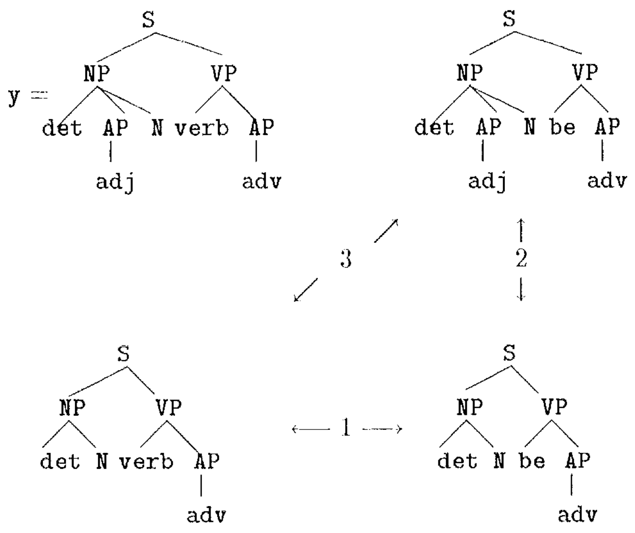


Figure 3: Analyses from the tree-bank and **y**, analysis of the prototype sentence obtained by analogy. Distances are given in nodes.

course, this naive solution implies an exponential explosion, but, fortunately, it is not necessary to consider the entire set of sentences, neither to generate them. Only sentences which are substrings of other strings may be coded. This allows us to use a simple non-deterministic version of the Aho & Corasick algorithm (Aho & Corasick 75), which only checks the possible presence of patterns on an array of integer triples. This algorithm competes well with one of the most efficient algorithm *agrep* (Wu & Manber 92), as it is faster in average.

4.1.3 Use and usefulness

In a first implementation, rather than really computing solutions of analogies on trees, we retrieve them from the tree-bank using approximate matching. Execution times are below one second for the analysis of short chunks of text (about 5 words). This technique helps a lot in the construction of tree-banks. Firstly, building new linguistic structures for new sentences is definitely made faster. Secondly, this technique enforces consistency, a sensible issue in tree-bank construction, especially if tree-banks are to be used to train probabilistic models.

4.1.4 Experiments and measures

To have a more precise idea about the power of the method, we carried out some experiments on an excerpt of the tree-bank of the University of Pennsylvania (787 sentences with their corresponding analyses). For all possible 4-tuples of sentences which verify the analogy definition, we computed the analysis of the first sentence by analogy. We recall that there may be no solution, one solution, or several solutions. As a restriction in this experiment, we did not consider distances between objects over half of the lengths of the objects.

Recall In document retrieval, *recall* is defined as the ratio of the number of relevant documents retrieved over the total number of relevant documents in the data base. Here, we define the *recall* as the number of times the exact structure was computed by analogy, divided by the number of sentence pairs having the same structure in the tree-bank. In our experiment, the recall is 0.69, a quite good figure, which shows that the technique is promising.

Precision Again, in document retrieval, *precision* is defined as the ratio of the number of relevant documents retrieved over the total number of documents retrieved. Here, we define the *precision* as the number of times when the exact structure was computed by analogy divided by the number of solutions delivered.

In the experiment, the precision is 0.43, which means that in almost half of the cases, one of the structures delivered is the right one. Now, in average, the structures delivered are far from the exact structure by 1.61 node, with a standard deviation of 1.86. This means that in average less that two nodes have to be edited in order to get the exact structure, the size of a structure in the tree-bank being 9.8 ± 5.4 nodes.

4.2 Generation by analogy

Generation may be performed in the same way as analysis, the difference being that the prototype is a tree and pattern-matching is performed on trees. The overall process is similar to the one for analysis, but in the opposite direction. The tool we have built for the edition of text with trees, allows approximate matching on trees, and generation is performed using the same functions as for analysis.

5 Conclusion

We have proposed a possible theoretical explanation of *analogy* in terms of edit distances. As expected, this proposal renders an account of some important linguistic phenomena, in particular, prefixing, suffixing and infixing. Also, transitivity is verified by linguistic examples. Nevertheless, the exact mathematical properties, and especially, the necessary and sufficient conditions on strings under which the above mentioned properties hold remain for the large part to be studied.

A possible application is analysis and generation by analogy. The proposed technique falls under the example-based approaches to natural language processing, but we think it may be safer than previous methods, because it relies on more information, and linguistically founded information. We have built a first implementation, which shows to be of great utility in accelerating the construction of tree-banks and improving their consistency.

References

Alfred V. Aho and Margaret J. Corasick
Efficient String Matching: An Aid to Bibliographic Search
Communications of the ACM, Vol. 18, No. 6, June 1975, pp. 333-340.

Gad M. Landau and Uzi Vishkin
Fast String Matching with k Differences
Journal of Computer and System Sciences, Vol. 37, 1988, pp. 63-78.

V.I. Levenshtein
Binary codes capable of correcting deletions, insertions and reversals
Dokl. Akad. Nauk SSSR, vol. 163, No. 4, August 1965, pp. 845-848.
English translation in *Soviet Physics-doklady* vol. 10, No. 8, February 1966, pp. 707-710.

Nagao Makoto
A Framework of a Mechanical Translation between Japanese and English by Analogy Principle
in *Artificial & Human Intelligence*,
Alick Elithorn and Ranan Banerji eds., Elsevier Science Publishers, NATO 1984.

Ron Y. Pinter
Efficient string matching with don't care patterns
in A. Apostolico and Z. Galil (eds) *NATO Series, vol. F12, Combinatorial Algorithms on Words*, Springer Verlag, Berlin Heidelberg, 1985, pp. 11-29.

Victor Sadler and Ronald Vendelmans
Pilot implementation of a bilingual knowledge bank
Proceedings of COLING-90, Helsinki, 1990, vol 3, pp. 449-451.

Sato Satoshi and Nagao Makoto
Example-Based Translation of Technical Terms
Proceedings of the Fifth International Conference on Theoretical and Methodological Issues in Machine Translation TMI-93, pp 58-68, Kyoto, 1993.

Ferdinand de Saussure
Cours de linguistique générale
publié par Charles Bally et Albert Sechehaye, Payot, Lausanne et Paris, 1916.

Stanley M. Selkow
The Tree-to-Tree Editing Problem
Information Processing Letters, Vol. 6, No. 6, December 1977, pp. 184-186.

Esko Ukkonen
On approximate string matching
in *Proc. Int. Conf. Found. Comp. Theor.*, Lecture Notes in Computer Science 158, Springer Verlag, Berlin/New York, 1983, pp 487-495.

Robert A. Wagner and Michael J. Fischer
The String-to-String Correction Problem
Journal for the Association of Computing Machinery, Vol. 21, No. 1, January 1974, pp. 168-173.

Sun Wu & Udi Manber
Fast Text Searching Allowing Errors
Communications of the ACM, Vol. 35, No. 10, October 1992, pp. 83-91.