Paradigmatic Cascades: a Linguistically Sound Model of Pronunciation by Analogy

François Yvon
ENST and CNRS, URA 820
Computer Science Department
46 rue Barrault - F 75 013 Paris
yvon@inf.enst.fr

Abstract

We present and experimentally evaluate a new model of pronunciation by analogy: the paradigmatic cascades model. Given a pronunciation lexicon, this algorithm first extracts the most productive paradigmatic mappings in the graphemic domain, and pairs them statistically with their correlate(s) in the phonemic domain. These mappings are used to search and retrieve in the lexical database the most promising analog of unseen words. We finally apply to the analogs pronunciation the correlated series of mappings in the phonemic domain to get the desired pronunciation.

1 Motivation

Psychological models of reading aloud traditionally assume the existence of two separate routes for converting print to sound: a direct lexical route, which is used to read familiar words, and a dual route relying upon abstract letter-to-sound rules to pronounce previously unseen words (Coltheart, 1978; Coltheart et al., 1993). This view has been challenged by a number of authors (e.g. (Glushsko, 1981)), who claim that the pronunciation process of every word, familiar or unknown, could be accounted for in a unified framework. These single-route models crucially suggest that the pronunciation of unknown words results from the parallel activation of similar lexical items (the *lexical neighbours*). This idea has been tentatively implemented both into various symbolic analogy-based algorithms (e.g. (Dedina and Nusbaum, 1991; Sullivan and Damper, 1992)) and into connectionist pronunciation devices (e.g. (Seidenberg and McClelland, 1989)).

The basic idea of these analogy-based models is to pronounce an unknown word x by recombining pronunciations of lexical items sharing common subparts with x. To illustrate this strategy, Dedina and Nussbaum show how the pronunciation of the sequence *lop* in the pseudo-word *blope* is analogized with the pronunciation of the same sequence in *sloping*. As there exists more than one way to recombine segments of lexical items, Dedina and Nussbaum's algorithm favors recombinations including *large* substrings of existing words. In this model, the similarity between two words is thus implicitely defined as a function of the length of their common subparts: the longer the common part, the better the analogy.

This conception of analogical processes has an important consequence: it offers, as Damper and Eastmond ((Damper and Eastmond, 1996)) state it, "no principled way of deciding the orthographic neighbours of a novel word which are deemed to influence its pronunciation (...)". For example, in the model proposed by Dedina and Nusbaum, any word having a common orthographic substring with the unknown word is likely to contribute to its pronunciation, which increases the number of lexical neighbours far beyond acceptable limits (in the case of *blope*, this neighbourhood would contain every English word starting in *bl*, or ending in *ope*, etc).

From a computational standpoint, implementing the recombination strategy requires a one-to-one alignment between the lexical graphemic and phonemic representations, where each grapheme is matched with the corresponding phoneme (a null symbol is used to account for the cases where the lengths of these representations differ). This alignment makes it possible to retrieve, for any graphemic substring of a given lexical item, the corresponding phonemic string, at the cost however of an unmotivated complexification of lexical representations.

In comparison, the paradigmatic cascades model (PCP for short) promotes an alternative view of analogical processes, which relies upon a linguistically motivated similarity measure between words.

The basic idea of our model is to take advantage of the internal structure of "natural" lexicons. In fact, a lexicon is a very complex object, whose elements are intimately tied together by a number of fine-grained relationships (typically induced by morphological processes), and whose content is severely restricted, on a language-dependant basis, by a complex of graphotactic, phonotactic and morphotactic constraints. Following e.g. (Pirrelli and Federici, 1994), we assume that these constraints surface simultaneously in the orthographical and in the phonological domain in the recurring pattern of *paradigmatically alterning* pairs of lexical items. Extending the idea originally proposed in (Federici, Pirrelli, and Yvon, 1995), we show that it is possible to extract these alternation patterns, to associate alternations in one domain with the related alternation in the other domain, and to construct, using this pairing, a fairly reliable pronunciation procedure.

2 The Paradigmatic Cascades Model

In this section, we introduce the paradigmatic cascades model. We first formalize the concept of a paradigmatic relationship. We then go through the details of the learning procedure, which essentially consists in an extensive search for such relationships. We finally explain how these patterns are used in the pronunciation procedure.

2.1 Paradigmatic Relationships and Alternations

The paradigmatic cascades model crucially relies upon the existence of numerous *paradigmatic relationships* in lexical databases. A paradigmatic relationship involves four lexical entries a, b, c, d, and expresses that these forms are involved in an analogical (in the Saussurian (de Saussure, 1916) sense) proportion: a is to b as c is to d (further along abbreviated as $a : b = c : d$, see also (Lepage and Shin-Ichi, 1996) for another utilization of this kind of proportions). Morphologically related pairs provide us with numerous examples of orthographical proportions, as in:

$$reactor : reaction = factor : faction \qquad (1)$$

Considering these proportions in terms of *orthographical alternations*, that is in terms of partial functions in the graphemic domain, we can see that each proportion involves two alternations. The first one transforms *reactor* into *reaction* (and *factor* into *faction*), and consists in exchanging the suffixes *or* and *ion*. The second one transforms *reactor* into *factor* (and *reaction* into *faction*), and consists in exchanging the prefixes *re* and f. These alternations are represented on figure 1.

$$\begin{tikzcd}
\text{reactor} \arrow[r, "f"] \arrow[d, "g"'] & \text{reaction} \arrow[d, "g"] \\
\text{factor} \arrow[r, "f"'] & \text{faction}
\end{tikzcd}$$

Figure 1: An Analogical Proportion

Formally, we define the notion of a paradigmatic relationship as follows. Given Σ, a finite alphabet, and $\mathcal{L}$, a finite subset of Σ^*, we say that $(a, b) \in \mathcal{L} \times \mathcal{L}$ is paradigmatically related to $(c, d) \in \mathcal{L} \times \mathcal{L}$ iff there exits two partial functions f and g from Σ^* to Σ^*, where f exchanges prefixes and g exchanges suffixes, and:

$$f(a) = c \quad \text{and} \quad f(b) = d \qquad (2)$$

$$g(a) = b \quad \text{and} \quad g(c) = d \qquad (3)$$

f and g are termed the *paradigmatic alternations* associated with the relationship $a : b =_{f,g} c : d$. The domain of an alternation f will be denoted by $dom(f)$.

2.2 The Learning Procedure

The main purpose of the learning procedure is to extract from a pronunciation lexicon, presumably structured by multiple paradigmatic relationships, the most productive paradigmatic alternations.

Let us start with some notations: Given G a graphemic alphabet and P a phonetic alphabet, a pronunciation lexicon $\mathcal{L}$ is a subset of $G^* \times P^*$. The restriction of $\mathcal{L}$ on G^* (respectively P^*) will be noted $\mathcal{L}_G$ (resp. $\mathcal{L}_P$). Given two strings x and y, $pref(x, y)$ (resp. $suff(x, y)$) denotes their longest common prefix (resp. suffix). For two strings x and y having a non-empty common prefix (resp. suffix) u, f^s_{xy} (resp, g^p_{xy}) denotes the function which transforms x into y: as $x = uv$, and as $y = ut$, f^s_{xy} substitutes a final v with a final t. λ denotes the empty string.

Given $\mathcal{L}$, the learning procedure searches $\mathcal{L}_G$ for any for every 4-uples (a, b, c, d) of graphemic strings such that $a : b =_{f,g} c : d$. Each match increments the productivity of the related alternations f and g. This search is performed using using a slightly modified version of the algorithm presented in (Federici, Pirrelli, and Yvon, 1995), which applies to every word x in $\mathcal{L}_G$ the procedure detailled in table 1.

In fact, the properties of paradigmatic relationships, notably their symetry, allow to reduce dramatically the cost of this procedure, since not all

```
GetAlternations(x)
$D(x) \leftarrow \{y \in \mathcal{L}_G/(t = pref(x,y)) \neq \lambda\}$
for $y \in D(x)$
do
  $P(x,y) \leftarrow \{(z,t) \in \mathcal{L}_G \times \mathcal{L}_G/z = f^s_{xy}(t)\}$
  if $P(x,y) \neq \emptyset$
    then
          $IncrementCount(f^s_{xy})$
          $IncrementCount(f^p_{xt})$
```

Table 1: The Learning Procedure

alternation			Example
x	→	x-/liː/	*good*
x-/t/	→	x-/ədliː/	*marked*
x-/əl/	→	x-/əliː/	*equal*
x-/əl/	→	x-/liː/	*capable*
x	→	x-/iː/	*cool*
x-/iːn/	→	x-/enliː/	*clean*
x-/ɪd/	→	x-/aɪdliː/	*id*
x-/ɪv/	→	x-/aɪliː/	*live*
x-/θ/	→	x-/ðliː/	*loath*
x	→	x-/laɪ/	*imp*
x-/ɪr/	→	x-/ɜːliː/	*ear*
x-/ɔn/	→	x-/onliː/	*on*

Table 2: Phonemic correlates of $x \rightarrow x - ly$

4-uple of strings in $\mathcal{L}_G$ need to be examined during that stage.

For each graphemic alternation, we also record their correlated alternation(s) in the phonological domain, and accordingly increment their productivity. For instance, assuming that *factor* and *reactor* respectively receive the pronunciations /fæktər/ and /riːæktər/, the discovery of the relationship expressed in (1) will lead our algorithm to record that the graphemic alternation $f \rightarrow re$ correlates in the phonemic domain with the alternation /f/ → /riː/. Note that the discovery of phonemic correlates does not require any sort of alignment between the orthographic and the phonemic representations: the procedure simply records the changes in the phonemic domain when the alternation applies in the graphemic domain.

At the end of the learning stage, we have in hand a set $\mathcal{A} = \{A_i\}$ of functions exchanging suffixes or prefixes in the graphemic domain, and for each A_i in $\mathcal{A}$:

(i) a statistical measure p_i of its productivity, defined as the likelihood that the transform of a lexical item be another lexical item:

$$p_i = \frac{\mid \{x \in dom(A_i) \text{ and } A_i(x) \in \mathcal{L}\} \mid}{\mid dom(A_i) \mid} \qquad (4)$$

(ii) a set $\{B_{i,j}\}, j \in \{1 \ldots n_i\}$ of correlated functions in the phonemic domain, and a statistical measure $p_{i,j}$ of their conditional productivity, i.e. of the likelihood that the phonetic alternation $B_{i,j}$ correlates with A_i.

Table 2 gives the list of the phonological correlates of the alternation which consists in adding the suffix *ly*, corresponding to a productive rule for deriving adverbs from adjectives in English. If the first lines of table 2 are indeed "true" phonemic correlates of the derivation, corresponding to various classes of adjectives, a careful examination of the last lines reveals that the extraction procedure is easily fooled by accidental pairs like *imp-imply*, *on-only* or *ear-early*. A simple pruning rule was used to get rid of these alternations on the basis of their productivity, and only alternations which were observed at least twice were retained.

It is important to realize that $\mathcal{A}$ allows to specifiy lexical neighbourhoods in $\mathcal{L}_G$: given a lexical entry x, its nearest neighbour is simply $f(x)$, where f is the most productive alternation applying to x. Lexical neighbourhoods in the paradigmatic cascades model are thus defined with respect to the locally most productive alternations. As a consequence, the definition of neighbourhoods implicitely incorporates a great deal of linguistic knowledge extracted from the lexicon, especially regarding morphological processes and phonotactic constraints, which makes it much for relevant for grounding the notion of analogy between lexical items than, say, any neighbourhood based on the string edition metric.

2.3 The Pronunciation of Unknown Words

Supose now that we wish to infer the pronunciation of a word x, which does not appear in the lexicon. This goal is achieved by exploring the neighbourhood of x defined by $\mathcal{A}$, in order to find one or several analogous lexical entry(ies) y. The second stage of the pronunciation procedure is to adapt the known pronunciation of y, and derive a suitable pronunciation for x: the idea here is to mirror in the phonemic domain the series of alternations which transform x into y in the graphemic domain, using the statistical pairing between alternations that is extracted during the learning stage. The complete pronunciation procedure is represented on figure 2.

Let us examine carefully how these two aspects of the pronunciation procedure are implemented. The first stage is to find a lexical entry in the neighbour-

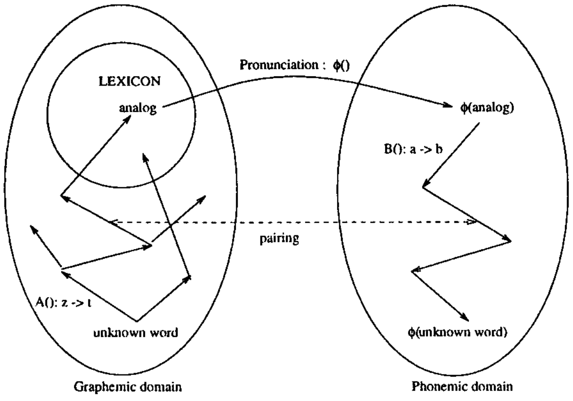

Figure 2: The pronunciation of an unknown word

hood of x defined by $\mathcal{L}$.

The basic idea is to generate $\mathcal{A}(x)$, defined as $\{A_i(x), for A_i \in \mathcal{A},\, x \in domain(A_i)\}$, which contains all the words that can be derived from x using a function in $\mathcal{A}$. This set, better viewed as a stack, is ordered according to the productivity of the A_i: the topmost element in the stack is the nearest neighbour of x, etc. The first lexical item found in $\mathcal{A}(x)$ is the analog of x. If $\mathcal{A}(x)$ does not contain any known word, we iterate the procedure, using x', the top-ranked element of $\mathcal{A}(x)$, instead of x. This expands the set of possible analogs, which is accordingly reordered, etc. This basic search strategy, which amounts to the exploration of a derivation tree, is extremely ressource consuming (every expension stage typically adds about a hundred of new virtual analogs), and is, in theory, not guaranted to terminate. In fact, the search problem is equivalent to the problem of parsing with an unrestricted Phrase Structure Grammar, which is known to be undecidable.

We have evaluated two different search strategies, which implement various ways to alternate between *expansion stages* (the stack is expanded by generating the derivatives of the topmost element) and *matching stages* (elements in the stack are looked for in the lexicon). The first strategy implements a depth-first search of the analog set: each time the topmost element of the stack is searched, but not found, in the lexicon, its derivatives are immediately generated, and added to the stack. In this approach, the position of an analog in the stack is assessed as a function of the "distance" between the original word x and the analog $y = A_{i_k}(A_{i_{k-1}}(\ldots A_{i_1}(x)))$, according to:

$$d(x,y) = \prod_{l=1}^{l=k} p_{i_l} \qquad (5)$$

The search procedure is stopped as soon an analog is found in $\mathcal{L}_G$, or else, when the distance between x and the topmost element of the stack, which monotonously decreases ($\forall i, p_i < 1$), falls below a pre-defined theshold.

The second strategy implements a kind of compromise between depth-first and breadth-first exploration of the derivation tree, and is best understood if we first look at a concrete example. Most alternations substituting one initial consonant are very productive, in English like in many other languages. Therefore, a word starting with say, a p, is very likely to have a very close derivative where the initial p has been replaced by say, a r. Now suppose that this word starts with *pl*: the alternation will derive an analog starting with *rl*, and will assess it with a very high score. This analog will, in turn, derive many more virtual analogs starting with *rl*, once its suffixes will have been substituted during another expansion phase. This should be avoided, since there are in fact very few words starting with the prefix *rl*: we would therefore like these words to be very poorly ranked. The second search strategy has been devised precisely to cope with this problem. The idea is to rank the stack of analogs according to the expectation of the number of lexical derivatives a given analog may have. This expectation is computed by summing up the productivities of all the alternations that can be applied to an analog y according to:

$$\sum_{i/y \in dom(A_i)} p_i \qquad (6)$$

This ranking will necessarily assess any analog starting in *rl* with a low score, as very few alternations will substitute its prefix. However, the computation of (6) is much more complex than (5), since it requires to examine a given derivative *before* it can be positioned in the stack. This led us to bring forward the lexical matching stage: during the expansion of the topmost stack element, all its derivatives are looked for in the lexicon. If several derivatives are simultaneously found, the search procedure halts and *returns more than one analog*.

The expectation (6) does not decrease as more derivatives are added to the stack; consequently, it cannot be used to define a stopping criterion. The search procedure is therefore stopped when all derivatives up to a given depth (2 in our experiments) have been generated, and unsuccessfully looked for in the lexicon. This termination criterion is very restrictive, in comparison to the one implemented in the depth-first strategy, since it makes it impossible to pronounce very long derivatives, for which a significant number of alternations need to be applied before an analog is found. An example is the word *synergistically*, for which the "breadth-first" search terminates uncessfully, whereas the depth-first search manages to retrieve the "analog" *energy*. Nonetheless, the results reported hereafter have been obtained using this "breadth-first" strategy, mainly because this search was associated with a more efficient procedure for reconstructing pronunciations (see below).

Various pruning procedures have also been implemented in order to control the exponential growth of the stack. For example, one pruning procedure detects the most obvious derivation cycles, which generate in loops the same derivatives; another pruning procedure tries to detect commutating alternations: substituting the prefix p, and then the suffix s often produces the same analog than when alternations apply in the reverse order, etc. More details regarding implementational aspects are given in (Yvon, 1996b).

If the search procedure returns an analog $y = A_{i_k}(A_{i_{k-1}}(\ldots A_{i_1}(x)))$ in $\mathcal{L}$, we can build a pronunciation for x, using the known pronunciation $\phi(y)$ of y. For this purpose, we will use our knowledge of the $B_{i,j}$, for $i \in \{i_1 \ldots i_k\}$, and generate every possible transforms of $\phi(y)$ in the phonological domain: $\{B^{-1}_{i_k,j_k}(B^{-1}_{i_{k-1},j_{k-1}}(\ldots(\phi(y))))\}$, with j_k in $\{1 \ldots n_{i_k}\}$, and order this set using some function of the $p_{i,j}$. The top-ranked element in this set is the pronunciation of x. Of course, when the search fails, this procedure fails to propose any pronunciation.

In fact, the results reported hereafter use a slightly extended version of this procedure, where the pronunciations of more than one analog are used for generating and selecting the pronunciation of the unknown word. The reason for using multiple analogs is twofold: first, it obviates the risk of being wrongly influenced by one very exceptional analog; second, it enables us to model *conspiracy effects* more accurately. Psychological models of reading aloud indeed assume that the pronunciation of an unknown word is not influenced by just one analog, but rather by its entire lexical neighbourhood.

3 Experimental Results

3.1 Experimental Design

We have evaluated this algorithm on two different pronunciation tasks. The first experiment consists in infering the pronunciation of the 70 pseudo-words originally used in Glushko's experiments, which have been used as a test-bed for various other pronunciation algorithms, and allow for a fair head-to-head comparison between the paradigmatic cascades model and other analogy-based procedures. For this experiment, we have used the entire nettalk (Sejnowski and Rosenberg, 1987) database (about 20 000 words) as the learning set.

The second series of experiments is intended to provide a more realistic evaluation of our model in the task of pronouncing unknown words. We have used the following experimental design: 10 pairs of disjoint (learning set, test set) are randomly selected from the nettalk database and evaluated. In each experiment, the test set contains about the tenth of the available data. A transcription is judged to be correct when it matches exactly the pronunciation listed in the database at the segmental level. The number of correct phonemes in a transcription is computed on the basis of the string-to-string edit distance with the target pronunciation. For each experiment, we measure the percentage of phoneme and words that are correctly predicted (referred to as correctness), and two additional figures, which are usually not significant in context of the evaluation of transcription systems. Recall that our algorithm, unlike many other pronunciation algorithms, is likely to remain silent. In order to take this aspect into account, we measure in each experiment the number of words that can not be pronounced at all (the silence), and the percentage of phonemes and words that are correctly transcribed *amongst those words that have been pronounced at all* (the precision). The average values for these measures are reported hereafter.

3.2 Pseudo-words

All but one pseudo-words of Glushko's test set could be pronounced by the paradigmatic cascades algorithm, and amongst the 69 pronunciation suggested by our program, only 9 were uncorrect (that is, were not proposed by human subjects in Glushko's experiments), yielding an overall correctness of 85.7%, and a precision of 87.3%.

An important property of our algortihm is that it allows to precisely identify, for each pseudo-word, the lexical entries that have been analogized, i.e. whose pronunciation was used in the inferential process. Looking at these analogs, it appears that three of our errors are grounded on very sensible analogies, and provide us with pronunciations that seem at least plausible, even if they were not suggested in Glushko's experiments. These were *pild* and *bild*, analogized with *wild*, and *pomb*, analogized with *tomb*.

These results compare favorably well with the performances reported for other pronunciation by analogy algorithms ((Damper and Eastmond, 1996) reports very similar correctness figures), especially if one remembers that our results have been obtained, *without resorting to any kind of pre-alignment between the graphemic and phonemic strings in the lexicons*.

3.3 Lexical Entries

This second series of experiment is intended to provide us with more realistic evaluations of the paradigmatic cascade model: Glushko's pseudowords have been built by substituting the initial consonant of existing monosyllabic words, and constitute therefore an over-simplistic test-bed. The nettalk dataset contains plurisyllabic words, complex derivatives, loan words, etc, and allows to test the ability of our model to learn complex morphophonological phenomenas, notably vocalic alternations and other kinds of phonologically conditioned root allomorphy, that are very difficult to learn.

With this new test set, the overall performances of our algorithm averages at about 54.5% of entirely correct words, corresponding to a 76% per phoneme correctness. If we keep the words that could not be pronounced at all (about 15% of the test set) apart from the evaluation, the per word and per phoneme precision improve considerably, reaching respectively 65% and 93%. Again, these precision results compare relatively well with the results achieved on the same corpus using other self-learning algorithms for grapheme-to-phoneme transcription (e.g. (van den Bosch and Daelemans, 1993; Yvon, 1996a)), which, unlike ours, benefit from the knowledge of the alignment between graphemic and phonemic strings. Table 3 summaries the performance (in terms of per word correctness, silence, and precision) of various other pronunciation systems, namely PRONOUNCE (Dedina and Nusbaum, 1991), DEC (Torkolla, 1993), SMPA (Yvon, 1996a). All these models have been tested using exactly the same evaluation procedure and data (see (Yvon, 1996b), which also contains an evalution performed with a French database suggesting that this learning strategy effectively applies to other languages).

System	corr.	prec.	silence
DEC	56.67	56.67	0
SMPA	63.96	64.24	0.42
PRONOUNCE	56.56	56.75	0.32
PCP	54.49	63.95	14.80

Table 3: A Comparative Evaluation

Table 3 pinpoints the main weakness of our model, that is, its significant silence rate. The careful examination of the words that cannot be pronounced reveals that they are either loan words, which are very isolated in an English lexicon, and for which no analog can be found; or complex morphological derivatives for which the search procedure is stopped before the existing analog(s) can be reached. Typical examples are: ***synergistically***, ***timpani***, ***hangdog***, ***oasis***, ***pemmican***, to list just a few. This suggests that the words which were not pronounced are not randomly distributed. Instead, they mostly belong to a linguistically homogeneous group, the group of foreign words, which, for lack of better evidence, should better be left silent, or processed by another pronunciation procedure (for example a rule-based system (Coker, Church, and Liberman, 1990)), than uncorrectly analogized.

Some complementary results finally need to be mentioned here, in relation to the size of lexical neighbourhoods. In fact, one of our main goal was to define in a sensible way the concept of a lexical neighbourhood: it is therefore important to check that our model manages to keep this neighbourhood relatively small. Indeed, if this neighbourhood can be quite large (typically 50 analogs) for short words, the number of analogs used in a pronunciation averages at about 9.5, which proves that our definition of a lexical neighbourhood is sufficiently restrictive.

4 Discussion and Perspectives

4.1 Related works

A large number of procedures aiming at the automatic discovery of pronunciation "rules" have been proposed over the past few years: connectionist models (e.g. (Sejnowski and Rosenberg, 1987)), traditional symbolic machine learning techniques (induction of decision trees, k-nearest neighbours) e.g. (Torkolla, 1993; van den Bosch and Daelemans, 1993), as well as various recombination techniques (Yvon, 1996a). In these models, orthographical correspondances are primarily viewed as resulting from a strict underlying *phonographical system*, where each grapheme encodes exactly one phoneme. This assumption is reflected by the possibility of aligning on a one-to-one basis graphemic and phonemic strings, and these models indeed use this kind of alignment to initiate learning. Under this view, the orthographical representation of individual words is strongly subject to their phonological forms on an word per word basis. The main task of a machine-learning algorithm is thus mainly to retrieve, on a statistical basis, these grapheme-phoneme correspondances, which are, in languages like French or

English, accidentally obscured by a multitude of exceptional and idiosyncratic correspondances. There exists undoubtly strong historical evidences supporting the view that the orthographical system of most european languages developped from a such phonographical system, and languages like Spanish or Italian still offer examples of that kind of very regular organization.

Our model, which extends the proposals of (Coker, Church, and Liberman, 1990), and more recently, of (Federici, Pirrelli, and Yvon, 1995), entertains a different view of orthographical systems. Even we if acknowledge the mostly phonographical organization of say, French orthography, we believe that the multiple deviations from a strict grapheme-phoneme correspondance are best captured in a model which weakens somehow the assumption of a strong dependancy between orthographical and phonological representations. In our model, each domain has its own organization, which is represented in the form of systematic (paradigmatic) set of oppositions and alternations. In both domain however, this organization is *subject to the same paradigmatic principle*, which makes it possible to represent the relationships between orthographical and phonological representations in the form of a statistical pairing between alternations. Using this model, it becomes possible to predict correctly the outcome in the phonological domain of a given derivation in the orthographic domain, including patterns of vocalic alternations, which are notoriously difficult to model using a "rule-based" approach.

4.2 Achievements

The paradigmatic cascades model offers an original and new framework for extracting information from large corpora. In the particular context of grapheme-to-phoneme transcription, it provides us with a more satisfying model of pronunciation by analogy, which:

- gives a principled way to automatically learn local similarities that implicitely incorporate a substantial knowledge of the morphological processes and of the phonotactic constraints, both in the graphemic and the phonemic domain. This has allowed us to precisely define and identify the content of lexical neighbourhoods;
- achieves a very high precision without resorting to pre-aligned data, and detects automatically those words that are potentially the most difficult to pronounce (especially foreign words). Interestingly, the ability of our model to process data which are not aligned makes it directly applicable to the reverse problem, i.e. phoneme-to-grapheme conversion.
- is computationally tractable, even if extremely ressource-consuming in the current version of our algorithm. The main trouble here comes from isolated words: for these words, the search procedure wastes a lot of time examining a very large number of very unlikely analogs, before realizing that there is no acceptable lexical neighbour. This aspect definitely needs to be improved. We intend to explore several directions to improve this search: one possibility is to use a graphotactical model (e.g. a n-gram model) in order to make the pruning of the derivation tree more effective. We expect such a model to bias the search in favor of short words, which are more represented than very long derivatives.

  Another possibility is to tag, during the learning stage, alternations with one or several morpho-syntactic labels expressing morphotactical restrictions: this would restrict the domain of an alternation to a certain class of words, and accordingly reduce the expansion of the analog set.

4.3 Perspectives

The paradigmatic cascades model achieves quite satisfactory generalization performances when evaluated in the task of pronouncing unknown words. Moreover, this model provides us with an effective way to define the lexical neighbourhood of a given word, on the basis of "surface" (orthographical) local similarities. It remains however to be seen how this model can be extended to take into account other factors which have been proven to influence analogical processes. For instance, frequency effects, which tend to favor the more frequent lexical neighbours, need to be properly model, if we wish to make a more realistic account of the human performance in the pronunciation task.

In a more general perspective, the notion of *similarity* between linguistic objects plays a central role in many corpus-based natural language processing applications. This is especially obvious in the context of example-based learning techniques, where the inference of some unknown linguistics property of a new object is performed on the basis of the most similar available example(s). The use of some kind of similarity measure has also demonstrated its effectiveness to circumvent the problem of data sparseness in the context of statistical language modeling.

In this context, we believe that our model, which is precisely capable of detecting local similarities in lexicons, and to perform, on the basis of these similarities, a global inferential transfer of knowledge, is especially well suited for a large range of NLP tasks. Encouraging results on the task of learning the English past-tense forms have already been reported in (Yvon, 1996b), and we intend to continue to test this model on various other potentially relevant applications, such as morpho-syntactical "guessing", part-of-speech tagging, etc.

References

Coker, Cecil H., Kenneth W. Church, and Mark Y. Liberman. 1990. Morphology and rhyming: two powerful alternatives to letter-to-sound rules. In *Proceedings of the ESCA Conference on Speech Synthesis*, Autrans, France.

Coltheart, Max. 1978. Lexical access in simple reading tasks. In G. Underwood, editor, *Strategies of information processing.* Academic Press, New York, pages 151–216.

Coltheart, Max, Brent Curtis, Paul Atkins, and Michael Haller. 1993. Models of reading aloud: dual route and parallel distributed processing approaches. *Psychological Review*, 100:589–608.

Damper, Robert I. and John F. G. Eastmond. 1996. Pronuncing text by analogy. In *Proceedings of the seventeenth International Conference on Computational Linguistics (COLING'96)*, pages 268–273, Copenhagen, Denmark.

de Saussure, Ferdinand. 1916. *Cours de Linguistique Générale.* Payot, Paris.

Dedina, Michael J. and Howard C. Nusbaum. 1991. PRONOUNCE: a program for pronunciation by analogy. *Computer Speech and Langage*, 5:55–64.

Federici, Stefano, Vito Pirrelli, and François Yvon. 1995. Advances in analogy-based learning: false friends and exceptional items in pronunciation by paradigm-driven analogy. In *Proceedings of IJ-CAI'95 workshop on 'New Approaches to Learning for Natural Language Processing'*, pages 158–163, Montréal.

Glushsko, J. R. 1981. Principles for pronouncing print: the psychology of phonography. In A. M. Lesgold and C. A. Perfetti, editors, *Interactive Processes in Reading*, pages 61–84, Hillsdale, New Jersey. Erlbaum.

Lepage, Yves and Ando Shin-Ichi. 1996. Saussurian analogy : A theoretical account and its application. In *Proceedings of the seventeenth International Conference on Computational Linguistics (COLING'96)*, pages 717–722, Copenhagen, Denmark.

Pirrelli, Vito and Stefano Federici. 1994. "Derivational" paradigms in morphonology. In *Proceedings of the sixteenth International Conference on Computational Linguistics (COLING'94)*, Kyoto, Japan.

Seidenberg, M. S. and James. L. McClelland. 1989. A distributed, developmental model of word recognition and naming. *Psychological review*, 96:523–568.

Sejnowski, Terrence J. and Charles R. Rosenberg. 1987. Parrallel network that learn to pronounce English text. *Complex Systems*, 1:145–168.

Sullivan, K.P.H and Robert I. Damper. 1992. Novel-word pronunciation within a text-to-speech system. In Gérard Bailly and Christian Benoit, editors, *Talking Machines*, pages 183–195. North Holland.

Torkolla, Kari. 1993. An efficient way to learn English grapheme-to-phoneme rules automatically. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 2, pages 199–202, Minneapolis, Apr.

van den Bosch, Antal and Walter Daelemans. 1993. Data-oriented methods for grapheme-to-phoneme conversion. In *Proceedings of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 45–53, Utrecht.

Yvon, François. 1996a. Grapheme-to-phoneme conversion using multiple unbounded overlapping chunks. In *Proceedings of the conference on New Methods in Natural Language Processing (NeMLaP II)*, pages 218–228, Ankara, Turkey.

Yvon, François. 1996b. *Prononcer par analogie : motivations, formalisations et évaluations.* Ph.D. thesis, École Nationale Supérieure des Télécommunications, Paris.