La /fɔnetizasjɔ̃/ comme un problème de translittération

Vincent Claveau
IRISA-CNRS
Campus de Beaulieu, 35042 Rennes cedex
vincent.claveau@irisa.fr

Résumé. La phonétisation est une étape essentielle pour le traitement de l'oral. Dans cet article, nous décrivons un système automatique de phonétisation de mots isolés qui est simple, portable et performant. Il repose sur une approche par apprentissage ; le système est donc construit à partir d'exemples de mots et de leur représentation phonétique. Nous utilisons pour cela une technique d'inférence de règles de réécriture initialement développée pour la translittération et la traduction. Pour évaluer les performances de notre approche, nous avons utilisé plusieurs jeux de données couvrant différentes langues et divers alphabets phonétiques, tirés du challenge Pascal Pronalsyl. Les très bons résultats obtenus égalent ou dépassent ceux des meilleurs systèmes de l'état de l'art.

Abstract. Phonetizing is a crucial step to process oral documents. In this paper, a new word-based phonetization approach is proposed ; it is automatic, simple, portable and efficient. It relies on machine learning ; thus, the system is built from examples of words with their phonetic representations. More precisely, it makes the most of a technique inferring rewriting rules initially developed for transliteration and translation. In order to evaluate the performances of this approach, we used several datasets from the Pronalsyl Pascal challenge, including different languages. The obtained results equal or outperform those of the best known systems.

Mots-clés : Phonétisation, phonémisation, inférence de règles de réécriture, challenge Pronalsyl, conversion graphème-phonème, translittération.

Keywords: Phonetization, phonemization, inference of rewriting rule, Pronalsyl challenge, grapheme-phoneme conversion, transliteration.

1 Introduction

La phonétisation est le processus qui associe à une séquence mots une ou plusieurs façons de la prononcer. C'est une étape essentielle pour le traitement de l'oral (transcription de la parole, synthèse de la parole, indexation de documents audios...). Les approches dictionnaire des premiers systèmes de traitement de l'oral ayant rapidement montré leurs limites, beaucoup ont cherché à développer des systèmes de phonétisation capables de manipuler des mots inconnus. Dans le contexte de la phonétisation de mots isolés à laquelle nous nous intéressons ici, l'approche la plus commune pour résoudre ce problème est de s'appuyer sur la forme graphique des mots pour deviner leur prononciation. En pratique, cela consiste à faire correspondre à un mot-forme (représenté par sa chaîne de caractère) une chaîne de symboles représentant une façon prototypique de prononcer ce mot-forme. C'est pourquoi cette tâche est aussi connue sous les noms de conversion lettre-phonème, conversion graphème-phonème, ou de phonémisation.

Dans notre cas, la phonétisation s'insère dans une problématique plus large d'indexation de flux vidéo dans laquelle nous sommes amenés à manipuler des mots isolés inconnus de notre système de transcription (néologismes, noms propres, sigles, imports de langues de spécialité). Pour les traiter, nous voulons disposer d'une technique produisant des phonétisations de bonne qualité, mais qui soit également rapide, automatique et portable pour pouvoir être adaptée à plusieurs langues et éventuellement à plusieurs sous-groupes de mots ou même à un locuteur. L'approche que nous proposons – que nous appelons IrisaPhon – répond à ces différents critères et part du constat que ce problème de phonétisation peut être vu comme de la translittération. Nous avons donc adapté une technique d'apprentissage que nous avions développée initialement pour la translittération et la traduction de termes biomédicaux (Claveau, 2007; Claveau, 2009). Cette technique permet d'inférer très efficacement des règles de réécriture à partir d'exemples, c'est-à-dire dans notre cas à partir de mots-formes couplés à leur représentation phonétique. Elle n'utilise aucune autre connaissance linguistique que ces exemples, assurant ainsi sa portabilité.

Après une revue des principales approches existantes en phonétisation, nous décrivons notre technique en section 3. Dans la section 4, nous présentons, comparons et discutons différents résultats d'évaluations. Quelques perspectives ouvertes par ce travail sont enfin présentées dans la dernière partie.

2 Travaux connexes

La phonétisation automatique de mot a déjà fait l'objet de nombreux travaux. La plupart adopte le paradigme de conversion lettre-phonème : la phonétisation comme séquence de phonèmes est déduite de la séquence de caractères formant le mot. Les techniques automatiques s'appuient sur des exemples de mots couplés à leur représentation phonétique. Ces exemples sont le plus généralement alignés lettre à lettre, souvent par des relations 1-1 (Black *et al.*, 1998; Damper *et al.*, 2005) mais de meilleures performances ont été obtenues en tenant compte d'alignements multiples (Bisani & Ney, 2002; Jiampojamarn *et al.*, 2007) qui rendent mieux compte du fait que plusieurs lettres peuvent être représentées par un phonème, et une lettre par plusieurs phonèmes. À partir de ces exemples, certains ont utilisé, et éventuellement adapté, des techniques d'apprentissage classiques comme les arbres de décision (Black *et al.*, 1998; Daelemans & Bosch, 1997) ou des techniques *lazy learning* (Bosch & Daelemans, 1998). D'autres ont mis l'emphase sur l'aspect séquentiel du problème et utilisent par exemple des HMM (Taylor, 2005) ou des techniques par analogie (Yvon, 1996; Marchand & Damper, 2000, *inter alia*). Enfin, certains ont tenté de mettre en œuvre des approches s'inspirant à la fois de l'apprentissage tout en tenant compte des aspects séquentiels. C'est le cas du système CSInf (2006) ou des approches de Jaimpojamarn *et al.* (2007; 2008) basés sur des SVM modifiés ou des HMM. Ces dernières approches qui intègrent bien les aspects séquentiels sont parmi les plus performantes. Nous revenons sur les performances de quelques-uns de ces systèmes dans la partie évaluation.

3 Phonétisation et réécriture

Comme nous l'avons annoncé précédemment, notre approche IrisaPhon prend ses racines dans un système d'apprentissage de règles de réécriture initialement développé pour la translittération de termes biomédicaux. Nous en rappelons les principes ci-dessous ; le lecteur intéressé peut se reporter à Claveau (2009) pour une description plus développée et son utilisation en traduction de termes biomédicaux.

Pour phonétiser un mot-forme inconnu, IrisaPhon lui applique des règles de réécriture et choisit la phonétisation la plus probable parmi les candidats générés à l'aide d'un modèle de langue. Les règles et le modèle de langue sont appris à partir de données d'entraînement, c'est-à-dire des listes de mots-formes (chaînes de caractères) couplés à leur représentation phonétique (chaînes de symboles phonétiques).

3.1 Apprentissage de règles de réécriture

La technique permettant d'inférer les règles de réécriture à partir des exemples est relativement simple. Une liste de mots couplés à leur représentation phonétique est donnée en entrée du système ; à chaque mot et représentation phonétique sont ajoutés deux caractères pour représenter le début et la fin de la chaîne de caractères (resp. # et $).

L'algorithme 1 décrit le processus qui permet d'inférer des règles à partir de cette liste d'exemples. La première étape, l'alignement, est réalisée à l'aide de DPalign (`http://www.cnts.ua.ac.be/~decadt/?section=dpalign`). Des caractères vides (notés '_') peuvent être insérés au besoin. Par la suite, le mot-forme en entrée (respectivement la phonétisation en sortie) d'une telle paire alignée p est noté $input(p)$ (resp. $output(p)$) ; de plus, $align(x, y)$ indique que la sous-chaîne x est alignée avec la sous-chaîne y dans la paire de termes considérée. Pour

Algorithme 1 Apprentissage des règles de réécriture

```
1: aligner les paires au niveau des lettres, mettre le résultat dans L
2: for all paire W_1 dans L do
3:    for all alignement de lettres dont les 2 lettres diffèrent dans W_1 do
4:       trouver la meilleure hypothèse de règles r dans l'espace de recherche E
5:       ajouter r à l'ensemble de règles R
6:    end for
7: end for
```

chaque différence entre deux lettres alignées, notre algorithme doit générer la règle de réécriture jugée la meilleure selon un certain score. Beaucoup de règles sont éligibles ; considérons par exemple la différence o/ɔ dans le couple #phonolog_y$ / #f_ɔnɑlɔdʒi$. Les règles o → ɔ, pho → f_ɔ, #phono → #f_ɔnɑ, etc., sont par exemple possibles.

Le score d'une règle est calculé sur la liste $\mathcal{L}$ comme le ratio entre le nombre de fois où la règle peut effectivement s'appliquer et le nombre de fois où la prémisse de la règle correspond à une sous-chaîne d'un mot-forme. Parmi toutes les règles possibles sur cet exemple, la règle maximisant ce score est donc retenue, et l'algorithme passe à une nouvelle différence entre l'*input* et l'*output* ou à un nouveau couple de $\mathcal{L}$.

La recherche de la meilleure règle parmi toutes celles possibles est l'étape clé de notre algorithme. Pour choisir cette règle dans notre espace de recherche de la manière la plus efficace possible, nous définissons une relation hiérarchique entre règles. Cette relation est notée par le symbole $\succeq$ (si $r_1 \succeq r_2$, alors r_1 est dite plus générale que r_2).

Définition 1 (Relation hiérarchique) *Soit r_1 et r_2 deux règles, alors* $r_1 \succeq r_2 \Leftrightarrow (input(r_1) \subseteq input(r_2) \wedge output(r_1) \subseteq output(r_2))$.

Cette relation est réflexive, transitive et anti-symétrique ; elle définit un ordre partiel sur l'espace de recherche $\mathcal{E}$ qui peut donc s'organiser sous forme de treillis. La figure 1 présente un extrait

du treillis de recherche construit à partir de la différence o/ə dans l'alignement #phonolog_y$ / #f_ənɑlədʒi$. En pratique, ces treillis sont explorés de haut en bas : les règles sont générées à

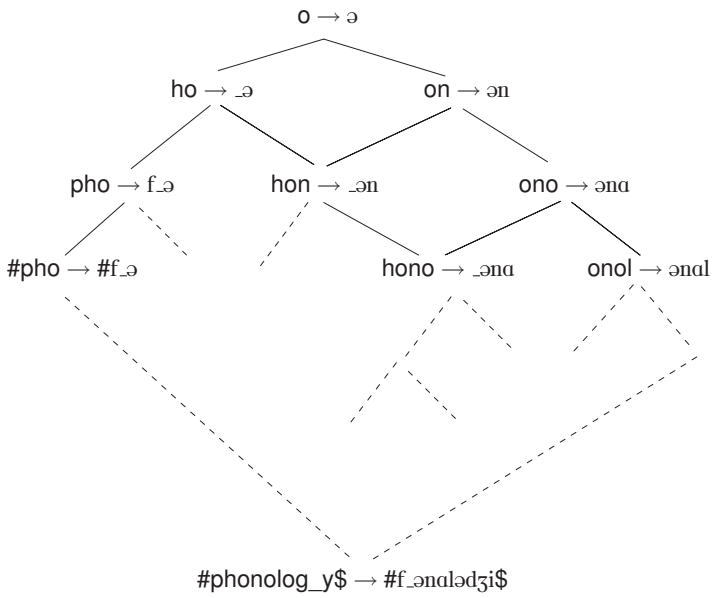

FIG. 1 – Treillis $\mathcal{E}$ de l'exemple o/ə dans #phonolog_y$ / #f_ənɑlədʒi$

la volée avec un opérateur très simple qui produit, pour une règle donnée, toutes les règles qui sont immédiatement plus spécifiques. En choisissant une fonction de score qui soit consistante avec cet opérateur de spécialisation et la structure de treillis qu'il sous-tend, il nous est possible de choisir rapidement la meilleure règle selon ce score (Claveau, 2009).

3.2 Choix de la phonétisation

Lorsqu'un mot nouveau doit être phonétisé, on lui applique toutes les règles de réécriture collectées, ce qui génère usuellement un grand nombre de phonétisations possibles. Il est important de noter que par construction, ces phonétisations sont alignées avec le mot de départ. Toutes ces alternatives sont conservées et la plus probable va être proposée. Cette probabilité est calculée de manière classique par un modèle de langue portant le couple mot/phonétisation. L'information de base (unigramme) de ce modèle de langue est donc une lettre alignée avec un symbole phonétique, que l'on note (par exemple) : $\begin{smallmatrix} \mathsf{s} \\ \mathsf{z} \end{smallmatrix}$. Avec les notations standard, pour un mot m aligné avec sa représentation phonétique f composés respectivement des lettres (y compris les vides _ ajoutés pour l'alignement) l_1, l_2, ..., l_m et k_1, k_2, ..., k_m, la probabilité se calcule par l'équation 1. En pratique, un historique de quelques lettres est suffisant. Dans les expériences présentées ci-dessous, cet historique est fixé à 6 lettres, et un lissage de Kneiser-Ney modifié est appliqué.

$$P\begin{pmatrix} m \\ f \end{pmatrix} = \prod_{i=1}^{m} P\left(\begin{matrix} l_i \\ k_i \end{matrix} \;\middle|\; \begin{matrix} l_1 \\ k_1 \end{matrix} , ..., \begin{matrix} l_{i-1} \\ k_{i-1} \end{matrix} \right) \tag{1}$$

Corpus	IrisaPhon	MIRA	M-M HMM	Joint n-gram	CSInf	PbA	LIA_PHON
Néerlandais CELEX	**95.58**	95.32	91.69	–	94.5	–	–
Allemand CELEX	93.60	**93.61**	90.31	92.5	–	–	–
Anglais NETtalk	**71.25**	67.82	59.32	64.6	–	65.35	–
Anglais CMUDict	**74.40**	71.99	65.38	–	–	–	–
Français Brulex	**94.75**	94.51	89.77	89.1	–	–	–

TAB. 1 – Précision en pourcentage d'IrisaPhon comparés à différents systèmes

4 Expérimentations

Pour évaluer notre approche, nous utilisons plusieurs jeux de données couvrant plusieurs langues et plusieurs jeux de phonèmes. Ces données sont celles proposées dans le cadre du *Letter-to-Phoneme Conversion Challenge* (Pronalsyl) du réseau Pascal `http://pascallin2.ecs.soton.ac.uk/Challenges/PRONALSYL`. Parmi les jeux de données disponibles, nous nous sommes concentrés sur ceux pour lesquels il existait des résultats publiés pour nous y comparer. Tous ces jeux de données comportent plusieurs milliers de paires réparties en 10 listes sur lesquelles les évaluations se font en validation croisée en 10 plis.

La mesure d'évaluation que nous utilisons est la précision en mot (moyennée sur les 10 tours de validation croisée) : nombre de mots parfaitement et entièrement phonétisés sur le nombre de mots donnés à phonétiser. C'est la mesure utilisée par les systèmes participant au challenge Pronalsyl.

Le tableau 1 présente la précision obtenue par IrisaPhon sur les différents jeux de données. À des fins de comparaison, nous indiquons également les résultats obtenus sur les mêmes jeux de données, lorsqu'ils sont disponibles, par différents systèmes de l'état de l'art. Ces systèmes sont : MIRA (Jiampojamarn *et al.*, 2008), M-M HMM (Jiampojamarn *et al.*, 2007), Joint n-gram (Demberg *et al.*, 2007), CSInf (Bosch & Canisius, 2006), PbA (Marchand & Damper, 2006), LIA_PHON (Béchet, 2001). Nous indiquons en gras les meilleurs résultats obtenus pour un jeu de test donné.

Le résultat est tout à fait satisfaisant puisqu'IrisaPhon obtient les meilleurs résultats sur quasiment tous les jeux de données. Il semble en particulier assez robuste aux jeux de données difficiles (NETtalk et CMUDict), bien qu'une large marge de progression subsiste.

5 Conclusion

La parti-pris de notre approche qui a été de considérer la phonétisation de mot comme un problème de translittération porte clairement ses fruits. Notre système IrisaPhon se compare avantageusement aux systèmes de l'état de l'art, aussi bien en terme de précision qu'en terme de temps de calcul. Bien sûr, les performances mesurées ici sur des données divisées artificiellement en jeu d'entraînement et jeu de test doivent être considérées comme des maxima, et des évaluations de notre système dans un contexte réel restent à mener. L'intégration de ce système dans notre problématique plus large d'indexation de document vidéo permettra de répondre en partie à ce soucis d'évaluation.

Références

BISANI M. & NEY H. (2002). Investigations on joint-multigram models for grapheme-to-phoneme conversion. In *Proceedings of the 7th International Conference on Spoken Language Processing*, Denver, USA.

BLACK A. W., LENZO K. & PAGEL V. (1998). Issues in building general letter to sound rules. In *Proceedings of the 3rd ESCA Workshop in Speech Synthesis*, Jenolan Caves, Australie.

BOSCH A. V. D. & CANISIUS S. (2006). Improved morpho-phonological sequence processing with constraint satisfaction inference. In *Proceedings of the 8th Meeting of the ACL Special Interest Group in Computational Phonology, SIGPHON'06*, p. 41–49, New York, USA.

BOSCH A. V. D. & DAELEMANS W. (1998). Do not forget: Full memory in memory-based learning of word pronunciation. In *Proceedings of NeMLaP3/CoNLL98*, Sydney, Australie.

BÉCHET F. (2001). LIA_PHON : un système complet de phonétisation de textes. *Traitement Automatique des Langues - TAL*, **42**(1), 47–67.

CLAVEAU V. (2007). Inférence de règles de réécriture pour la traduction de termes biomédicaux. In *Actes de la conférence Traitement automatique des langues naturelles, TALN'07*, Toulouse, France.

CLAVEAU V. (2009). Translation of biomedical terms by inferring rewriting rules. In V. PRINCE & M. ROCHE, Eds., *Information Retrieval in Biomedicine: Natural Language Processing for Knowledge Integration*. IGI - Global.

DAELEMANS W. & BOSCH A. V. D. (1997). Language-independent data-oriented grapheme-to-phoneme conversion. In *Progress in Speech Synthesis*, p. 77–89. New York, USA.

DAMPER R. I., MARCHAND Y., MARSTERS J. D. & BAZIN A. I. (2005). Aligning text and phonemes for speech technology applications using an em-like algorithm. *International Journal of Speech Technology*, **8**(2).

DEMBERG V., SCHMID H., & MÖHLER G. (2007). Phonological constraints and morphological preprocessing for grapheme-to-phoneme conversion. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, p. 96–103, Prague, République tchèque.

JIAMPOJAMARN S., CHERRY C. & KONDRAK G. (2008). Joint processing and discriminative training for letter-to-phoneme conversion. In *Proceedings of ACL HLT 2008*, Columbus, USA.

JIAMPOJAMARN S., KONDRAK G., & SHERIF T. (2007). Applying many-to-many alignments and hidden markov models to letter-to-phoneme conversion. In *Proceedings of the conference of the North American Chapter of the Association for Computational Linguistics*, Rochester, New York, USA.

MARCHAND Y. & DAMPER R. I. (2000). A multistrategy approach to improving pronunciation by analogy. *Computational Linguistics*, **26**(2).

MARCHAND Y. & DAMPER R. I. (2006). Can syllabification improve pronunciation by analogy of english ? *Natural Language Engineering*, **13**(1).

TAYLOR P. (2005). Hidden markov models for grapheme to phoneme conversion. In *Proceedings of the 9th European Conference on Speech Communication and Technology*, Lisbonne, Portugal.

YVON F. (1996). *Prononcer par analogie : motivations, formalisations et évaluations*. Thèse de doctorat, ENST, Paris.