Annotation syntaxique automatique de la partie orale du CÉFC

Alexis Nasr, Franck Dary, Frédéric Bechet, Benoit Favre

▶ **To cite this version:**

Alexis Nasr, Franck Dary, Frédéric Bechet, Benoit Favre. Annotation syntaxique automatique de la partie orale du CÉFC. Langages, 2020. hal-02973242

HAL Id: hal-02973242
https://hal.science/hal-02973242

Submitted on 6 Nov 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Alexis Nasr
Laboratoire Informatique et Systèmes (CNRS UMR 7020) & Aix-Marseille Université

Frédéric Béchet
Laboratoire Informatique et Systèmes (CNRS UMR 7020) & Aix-Marseille Université

Franck Dary
Laboratoire Informatique et Systèmes (CNRS UMR 7020) & Aix-Marseille Université

Benoit Favre
Laboratoire Informatique et Systèmes (CNRS UMR 7020) & Aix-Marseille Université

Annotation syntaxique automatique de la partie orale du CÉFC

1. INTRODUCTION

La quantité de données traitées dans le cadre du projet ORFÉO, de l'ordre de plus de trois millions de mots pour sa partie orale, rendait impossible une annotation manuelle de ces dernières. C'est la raison pour laquelle cette annotation a été réalisée de manière automatique, en acceptant l'idée qu'elle comporte des erreurs, pour autant que l'on soit capable d'en estimer la quantité et la nature.

Cet article décrit les logiciels qui ont été utilisés ainsi que leurs performances. Certains préexistaient au projet (Nasr *et al.* 2011) et ont été adaptés aux données d'ORFÉO tandis que d'autres ont été développés spécifiquement. Bien que cet article ne soit pas un article de traitement automatique des langues (TAL), il nous a semblé important de donner une description assez précise des outils sans toutefois entrer dans des détails non pertinents. Toujours du point de vue du TAL, ce travail présente deux originalités qui sont la prise en compte des pauses silencieuses et des informations décrivant les conditions de production des énoncés, dans les traitements linguistiques.

L'article est composé de deux parties. Dans la partie 2, on abordera la nature des annotations linguistiques qui ont été produites, puis les outils qui ont permis de les réaliser. Dans la partie 3, on décrira les résultats obtenus par ces outils. Les résultats que nous présentons ici portent exclusivement sur la partie orale du CÉFC.

2. TRAITEMENTS AUTOMATIQUES

Les traitements linguistiques des données orales d'ORFÉO consistent en six étapes. Leur point de départ est constitué des enregistrements audios et de leurs transcriptions. La première étape est l'alignement texte-son qui consiste à identifier, dans le signal audio, le début et la fin de chaque mot de la transcription. Cette étape est réalisée par le logiciel JTRANS (Cerisara, Mella & Fohr 2009) et ne sera pas décrite ici. La seconde consiste à segmenter les transcriptions en unités lexicales, puis à leur associer une partie du discours et un lemme. Ces opérations sont généralement appelées ‹segmentation lexicale›, ‹étiquetage en partie du discours› (ou POS) et ‹lemmatisation›. Les deux étapes suivantes sont l'analyse syntaxique et la segmentation en énoncés.

D'un point de vue formel, ces étapes peuvent être vues comme des opérations de segmentation, d'étiquetage et d'établissement de relations entre unités. La segmentation consiste à segmenter la chaîne orthographique. Dans notre cas, il s'agit de segmenter les transcriptions en unités lexicales et en énoncés. L'étiquetage consiste à associer aux unités une étiquette véhiculant une information linguistique. Pour ce faire, nous associons aux unités lexicales une partie du discours et un lemme. La mise en relation consiste à établir des liens entre des unités. Cette mise en relation consiste ici à établir des dépendances syntaxiques entre unités lexicales.

2.1. Format

Nous avons représenté, dans la Figure 1, le résultat de l'analyse d'un énoncé extrait du corpus TCOF au format ORFÉO.

Ce format est compatible avec celui des UNIVERSAL DEPENDENCIES (Nivre *et al.* 2016), lui-même inspiré du format des campagnes d'évaluation CONLL (*Computational Natural Language Learning*) (Nivre *et al.* 2007). À l'exception des deux premières, chaque ligne de l'exemple correspond à une unité lexicale.

Chaque unité lexicale est représentée dans la colonne 2 et un index lui est attribué en colonne 1. La colonne 3 correspond au lemme de l'unité lexicale et les colonnes 4 et 5 à sa partie du discours. La colonne 6 permet de représenter le résultat d'une analyse morphologique, elle n'est pas utilisée ici. Les deux colonnes 7 et 8 décrivent la structure syntaxique de l'énoncé. La colonne 7 indique la position, dans l'énoncé, du gouverneur syntaxique du mot courant. On voit, par exemple, que le gouverneur syntaxique du pronom *je*, en position 2, est le verbe *servir*, en position 5. La colonne 8 indique la fonction syntaxique du mot. Suivent deux colonnes non utilisées, elles ont été conservées pour des raisons de compatibilité avec le format des UNIVERSAL DEPENDENCIES. Les colonnes 11 et 12 correspondent au temps de début et au temps de fin de la prononciation du mot. La colonne 13 porte l'information du locuteur.

La première des deux lignes qui débutent par le caractère dièse est le nom du fichier dont provient l'énoncé et de sa position dans celui-ci. La seconde est une représentation horizontale de l'énoncé, qui facilite la recherche dans les fichiers.

```
# sent_id = cefc-tcof-Escalade_mic-10
# text = moi je s~ me sers du support euh de la structure ...
1  moi           moi         PRO PRO _ 5  periph   _ _ 46.38 46.55 L2
2  je            je          CLS CLS _ 5  subj     _ _ 46.56 47.05 L2
3  s~            _           XXX XXX  _ 2  disflink _ _ 47.06 47.11 L2
4  me            me          CLI CLI _ 5  dep      _ _ 47.86 48.78 L2
5  sers          servir      VRB VRB _ 0  ROOT     _ _ 48.79 49.08 L2
6  du            _           PRE PRE _ 5  dep      _ _ 49.13 49.20 L2
7  support       support     NOM NOM _ 6  dep      _ _ 49.30 49.71 L2
8  euh           euh         INT INT _ 7  disflink _ _ 49.72 49.74 L2
9  de            de          PRE PRE _ 7  dep      _ _ 50.25 50.53 L2
10 la            le          DET DET _ 11 spe      _ _ 50.58 50.82 L2
11 structure     structure   NOM NOM _ 9  dep      _ _ 50.83 51.13 L2
12 artificielle  artificiel ADJ ADJ _ 11 dep      _ _ 51.14 51.66 L2
13 d'            de          PRE PRE _ 11 dep      _ _ 51.67 51.69 L2
14 escalade      escalade    NOM NOM _ 13 dep      _ _ 51.70 52.21 L2
```

Figure 1 : Représentation au format ORFÉO de l'énoncé :
moi je s~ me sers du support euh de la structure artificielle d'escalade

2.2. Outils

Nous détaillons, dans cette section, les différents traitements effectués. À l'exception de la segmentation lexicale, tous les traitements s'intègrent dans un cadre général, appelé la ‹machine à états›. C'est la raison pour laquelle nous commençons par décrire rapidement la segmentation lexicale, puis nous décrirons la machine à états. La segmentation lexicale est réalisée à l'aide du lexique ORFÉO, décrit dans l'article de J. Deulofeu et A. Valli (2020) dans le présent numéro. Il s'agit d'un traitement simple, réalisé en deux étapes.

La première consiste en une segmentation fondée sur des règles typographiques, indépendantes d'un lexique. Ces règles consistent à découper la chaîne en se fondant sur les caractères d'espacement et de ponctuation. À l'issue de cette étape, le texte est découpé en unités lexicales minimales.

La seconde étape consiste à regrouper des unités lexicales minimales pour former des unités lexicales complexes. Ce traitement est réalisé à l'aide du lexique ORFÉO qui identifie 3 726 unités lexicales complexes, sur un total de 539 907. Lorsqu'une unité lexicale complexe apparaît dans le lexique, elle est reconnue comme telle dans la chaîne. Dans certains cas, les composantes de ces unités complexes ne sont pas autonomes, à l'image de la première composante de la conjonction de subordination complexe *parce que*. Le regroupement des unités lexicales simples au sein d'une unité complexe garantit alors une segmentation correcte. Dans d'autres cas, les composantes sont autonomes et le regroupement peut aboutir à une segmentation erronée, comme dans le cas de *pourvu que* pour lequel on peut imaginer un contexte dans lequel il ne constitue pas une unité complexe, par exemple : *il en était si bien pourvu qu'il ne craignait pas le froid*.

La règle principale qui a présidé à la création des unités complexes du lexique ORFÉO est qu'une suite d'unités minimales qui constitue, dans une très large mesure, une unité complexe, est ajoutée au lexique. Cette règle comporte une part subjective, dans la mesure où ce sont les créateurs du lexique qui ont jugé du caractère quasi systématique du regroupement. Dans certains cas, en particulier pour les catégories des prépositions et des conjonctions, une étude sur corpus (Ramisch *et al.* 2016 ; Nasr *et al.* 2015) a permis de quantifier la justesse de ce regroupement. Plus de détails sur les aspects linguistiques du problème sont évoqués dans l'article de J. Deulofeu et A. Valli (2020). Nous n'avons pas pour l'heure d'estimation du nombre d'erreurs provoquées par l'application de cette règle.

Comme nous l'avons mentionné *supra*, la segmentation lexicale constitue un traitement à part. Tous les autres traitements sont réalisés par un ensemble d'outils regroupés dans un cadre de la ‹machine à états›.

2.3. La machine à états

Le terme *machine à états* est emprunté à l'informatique théorique. Dans notre cas, il s'agit d'un programme qui lit les mots des énoncés et produit, pour le mot courant, une annotation linguistique. Chaque type d'annotation (partie de discours, lemme, etc.) correspond à un état de la machine. Une fois que l'annotation correspondant à l'état est effectuée, la machine passe à l'état suivant. L'exemple de la section précédente a été réorganisé dans la Figure 2 de manière à mieux expliquer le fonctionnement de la machine à états. Cette nouvelle organisation consiste principalement à modifier l'ordre des colonnes de manière à représenter, dans les quatre colonnes de gauche, les informations qui ne sont pas issues d'un traitement. La dernière colonne indique, pour chaque mot, s'il constitue la fin d'un énoncé. Finalement, la position du gouverneur d'un mot est donnée par sa position relativement à ce dernier, ce qui permet de se passer de l'indice des mots.

```
L2 46.38 46.55 moi          PRO  moi         4 periph   0
L2 46.56 47.05 je           CLS  je          3 subj     0
L2 47.06 47.11 s~           XXX  _          -1 disflink 0
L2 47.86 48.78 me           CLI  me          1 dep      0
L2 48.79 49.08 sers         VRB  servir      0 ROOT     0
L2 49.13 49.20 du           PRE  _          -1 dep      0
L2 49.30 49.71 support      NOM  support    -1 dep      0
L2 49.72 49.74 euh          INT  euh        -1 disflink 0
L2 50.25 50.53 de           PRE  de         -2 dep      0
L2 50.58 50.82 la           DET  le          1 spe      0
L2 50.83 51.13 structure    NOM  structure  -2 dep      0
L2 51.14 51.66 artificielle ADJ  artificiel -1 dep      0
L2 51.67 51.69 d'           PRE  de         -2 dep      0
L2 51.70 52.21 escalade     NOM  escalade   -1 dep      1
```

Figure 2 : Un exemple de sortie de la machine à états

Bien entendu, dans la réalité, ces colonnes sont issues d'un traitement, qu'il soit manuel, comme la transcription orthographique et l'identification du locuteur par exemple, ou automatique, comme l'alignement temporel. Les cinq colonnes de droite sont le résultat d'un traitement réalisé par la machine à états. Cette dernière possède, comme évoqué *supra*, des états qui correspondent aux différents traitements réalisés : un état pour l'étiquetage en partie du discours, un pour la lemmatisation et un pour l'analyse syntaxique. Elle possède, de plus, plusieurs bandes de lecture/écriture ainsi qu'une tête de lecture/écriture qui permet d'écrire sur chacune des bandes. Chaque bande correspond à une colonne dans la Figure 2.

À tout moment, la machine se trouve dans une ‹configuration› qui spécifie l'état de la machine, la position de la tête, le contenu des différentes bandes ainsi que le contenu d'une pile. Cette dernière garde en mémoire un certain nombre de mots de l'énoncé dont l'analyse syntaxique est en cours, soit que leur gouverneur syntaxique n'a pas encore été déterminé, soit que leurs dépendants n'ont pas tous été identifiés (cf. § 2.7).

La machine procède par étapes : à chaque étape, une information linguistique est prédite et écrite à la position courante de la tête, sur la bande correspondante. Puis, la tête est déplacée vers la droite et une transition est empruntée, permettant de changer l'état de la machine.

Chaque état de la machine repose sur un classifieur qui prédit une étiquette étant donné une configuration (cf. § 2.4).

Du point de vue du TAL, l'originalité de la machine à états est qu'elle permet, pour chaque prédiction, de prendre en compte l'ensemble des prédictions réalisées jusque-là, quelle que soit leur nature, comportement que l'on qualifie d'‹incrémental›. Ainsi, lors de la prédiction de la partie du discours d'un mot, la machine a accès aux mots de l'énoncé, aux parties du discours, aux lemmes, ainsi qu'à l'analyse syntaxique des mots précédents.

Ce comportement se distingue du traitement séquentiel, plus classique en TAL, où l'intégralité d'un texte subit séquentiellement les différents traitements. Pour reprendre l'exemple précédent de la prédiction de la partie du discours d'un mot, dans un traitement séquentiel, il n'est pas possible de prendre en compte les lemmes ou l'analyse syntaxique des mots précédents, pour la simple raison que les traitements correspondants n'ont pas encore été réalisés. En revanche, dans un traitement séquentiel, l'analyse syntaxique d'un mot a accès aux parties du discours des mots suivants, ce qui n'est pas possible dans un traitement incrémental. De manière schématique, dans un traitement séquentiel, la matrice de la Figure 2 est remplie colonne par colonne alors que, dans un traitement incrémental, elle est remplie ligne par ligne. La raison pour laquelle nous avons opté pour une architecture incrémentale dans le cadre du projet ORFÉO sera précisée *infra* (§ 2.7).

Dans les sections suivantes, nous décrirons tout d'abord le modèle de classification utilisé pour réaliser les différentes prédictions, puis nous détaillerons les instanciations de ce modèle permettant de réaliser les traitements.

2.4. Classifieur

Le modèle de classifieur utilisé dans la machine à états est le « perceptron multicouches » (Rumelhart *et al.* 1986), qui est un type de réseaux de neurones. Ce dernier prend en entrée la configuration dans laquelle se trouve la machine à états et produit une classe matérialisée par une étiquette. Cette dernière peut être de nature très diverse, elle peut correspondre à une partie du discours, à une règle morphologique ou à une action d'un analyseur syntaxique.

Comme nous l'avons vu *supra*, une configuration est un objet complexe, qui comporte en particulier le contenu de toutes les bandes. Seules certaines parties de ces informations, jugées pertinentes, sont fournies au classifieur. Ces informations sont spécifiées sous la forme d'un ‹modèle de traits›, qui précise les éléments de la configuration jugés pertinents. Formellement, ce modèle se présente sous la forme d'un ensemble de fonctions indicatrices, chacune d'entre elles permettant d'accéder à une information particulière de la configuration. Ces fonctions sont, en réalité, la composition de deux fonctions plus simples, une ‹fonction d'adresse›, qui spécifie un élément particulier de la configuration, par exemple le mot courant, et une ‹fonction d'attribut› qui spécifie un aspect de cet élément, par exemple sa partie du discours, son lemme, sa terminaison, son locuteur ou sa durée.

L'application d'une fonction indicatrice f à une configuration C renvoie l'information correspondant à f dans C. Cette information est représentée sous la forme d'un vecteur binaire qui a pour dimension le domaine de la fonction (le nombre de valeurs différentes qu'elle peut renvoyer). Toutes les composantes du vecteur valent zéro, à l'exception de celle qui correspond à la valeur de la fonction, qui vaut un. L'entrée du classifieur se présente alors sous la forme d'un grand vecteur binaire qui est la concaténation des vecteurs renvoyés par chacune des fonctions du modèle de traits. Il s'agit d'un moyen de représentation particulièrement souple car il permet de prendre en compte n'importe quelle information qui semble pertinente pour la tâche à réaliser par le classifieur. Nous verrons *infra* (§ 3) qu'il permet de prendre en compte des informations prosodiques, telles que la durée des silences entre mots, ou encore des métadonnées représentant les conditions de production des énoncés.

Comme souvent dans le cas des réseaux de neurones, les mots ne sont pas représentés sous la forme d'un vecteur binaire. La raison en est que, étant donné la taille du lexique, la représentation vectorielle est extrêmement coûteuse : la dimension des vecteurs est égale à la taille du lexique et seule une composante vaut 1. On préfère une représentation plus compacte, sous la forme de « plongements de mots » *(word embeddings)* (Mikolov *et al.* 2013) qui sont des vecteurs

continus (chaque composante du vecteur est un réel) de dimension inférieure à la taille du lexique de plusieurs ordres de grandeur.

Le classifieur produit en sortie un vecteur de nombres réels dont la dimension est égale au nombre de classes. Chaque composante de ce vecteur correspond au score attribué par le classifieur à la classe correspondant. La décision finale consiste à choisir la classe ayant obtenu le meilleur score.

2.5. Étiqueteur en parties du discours

L'étiqueteur en parties du discours prend en entrée une configuration de la machine et prédit une des 19 parties du discours du format d'annotation d'ORFÉO. Il s'agit d'une architecture simple dans la mesure où la classe produite correspond directement à l'information linguistique qui sera écrite sur les bandes de la machine.

Dans certains cas, il n'est pas possible d'attribuer une partie du discours à un mot, auquel cas on lui associe l'étiquette ‹XXX›, c'est notamment le cas pour les mots tronqués.

L'étiqueteur utilise aussi la ‹signature› des mots. Cette dernière est l'ensemble des parties du discours possibles d'un mot représenté sous la forme d'un vecteur binaire de dimension 19 (nombre de parties du discours) qui comporte la valeur 1 pour toutes les parties du discours possibles du mot. La prise en compte de la signature des formes interdit d'associer à ces dernières une partie du discours qu'elles ne possèdent pas dans le lexique. Les signatures sont calculées à partir du lexique ORFÉO qui comporte à peu près 540 000 formes fléchies. Lorsqu'une forme est absente du lexique, il lui est associé une signature correspondant aux différentes classes ouvertes, ce qui revient à dire que les formes inconnues ne pourront avoir comme partie du discours une classe fermée.

2.6. Lemmatiseur

La lemmatisation consiste à produire le lemme correspondant à une forme. Une telle tâche ne peut être vue directement comme un problème de classification car le nombre de classes est très important, potentiellement infini pour les classes ouvertes. En revanche, pour ce qui relève de la morphologie régulière, le nombre de règles de flexions est limité, de sorte que la conjugaison verbale, par exemple, peut être modélisée sous la forme d'une tâche de classification.

En pratique, le lemmatiseur procède en deux étapes. La première consiste à traiter les cas de morphologie non régulière, à l'aide d'un dictionnaire d'exceptions et, la seconde, les cas de morphologie régulière à l'aide de règles.

Lors de la première étape, un accès est réalisé à l'aide d'un couple composé d'une forme et d'une partie du discours au dictionnaire des exceptions. Ce dernier est un ensemble de triplets composés d'une forme, d'une partie du discours

et d'un lemme, par exemple [*soyons*, VNF, *être*], où VNF est l'étiquette correspondant à un verbe conjugué. Cette entrée indique que le lemme correspondant au couple [*soyons*, VNF] est le verbe *être*. Si le dictionnaire des exceptions possède une entrée pour le couple alors le lemme correspondant est renvoyé.

Lorsque l'accès au dictionnaire des exceptions échoue, le couple composé de la forme et de la partie du discours est donné en entrée à un classifieur qui prédit une règle. Cette dernière se présente sous la forme d'un couple [$-s_1$, $+ s_2$] dans lequel s_1 et s_2 sont des chaînes de caractères. Une telle règle indique qu'il faut retirer le suffixe s_1 à la forme et lui accoler le suffixe s_2 pour obtenir le lemme. Pour l'entrée [*partîmes*, VNF] par exemple, la règle [*-îmes,+ir*] est produite, qui, appliquée à la forme *partîmes*, produit le lemme *partir*.

Le dictionnaire d'exceptions est extrait de manière automatique du lexique ORFÉO en même temps que les règles de flexions. Le principe est simple, les règles de flexion sont d'abord extraites, par examen des suffixes, puis, les entrées ne correspondant pas à des règles sont ajoutées au lexique d'exception.

2.7. Analyseur syntaxique

L'analyseur syntaxique repose sur le modèle de l'analyse par transition (Yamada & Matsumoto 2003). Il s'agit d'une analyse syntaxique en dépendances, qui attribue à tout mot de l'énoncé une fonction et un gouverneur syntaxiques. Le texte est traité mot à mot ; pour tout mot lu, l'algorithme cherche à identifier son gouverneur et sa fonction. La difficulté provient du fait que le gouverneur d'un mot peut être arbitrairement éloigné de celui-ci. Afin de résoudre ce problème, l'analyse par transition utilise une mémoire dans laquelle sont stockés les mots à la recherche d'un dépendant ou d'un gouverneur. La mémoire prend la forme d'une pile, le mot se trouvant au sommet de cette dernière possède un statut particulier, c'est lui qui sera gouverneur ou dépendant de la prochaine dépendance prédite. À chaque étape, l'analyseur doit décider si une dépendance doit être établie entre le mot courant et le sommet de la pile. Deux choix sont alors possibles : soit le mot au sommet de la pile est gouverneur et le mot courant dépendant, soit l'inverse. L'analyseur peut aussi décider de ne pas établir de dépendance entre ces deux mots, auquel cas le mot courant est mis dans la pile, il en constitue alors le sommet.

Chacune de ces actions peut être vue comme une classe à prédire. Ce type d'analyse s'intègre donc parfaitement dans le cadre de la machine à états.

Il possède de plus un avantage décisif dans notre cas, qui est sa capacité à prédire à tout moment que la fin d'un énoncé a été identifiée. Il s'agit d'un avantage important dans la mesure où, en général, les analyseurs syntaxiques ont été développés pour des textes écrits dont on dispose de la ponctuation. Dans ce cas, la segmentation en énoncés est généralement réalisée avant l'analyse syntaxique. Dans le cas des transcriptions de l'oral, cette approche n'est pas viable car la ponctuation n'existe généralement pas et la structure syntaxique

aide à prédire les fins d'énoncés. C'est pourquoi, nous avons opté pour une architecture incrémentale qui permet, à tout moment, de décider si un mot constitue une fin d'énoncé, en fonction de l'analyse syntaxique précédente.

3. EXPÉRIENCES ET RÉSULTATS

3.1. Données

La machine à états repose sur des classifieurs dont le comportement doit être appris à partir de données pour lesquelles on dispose d'une annotation linguistique correcte. Pour cela, un certain nombre de fichiers du CÉFC ont été sélectionnés et annotés manuellement. Chaque fichier a été segmenté en énoncés et chaque mot s'est vu attribuer une partie du discours, un lemme, un gouverneur et une fonction syntaxique. Ces annotations ont été réalisées en conformité avec les principes d'annotation du projet, consignés dans les guides d'annotation. L'ensemble de ces données constitue le corpus de référence du CÉFC, il est constitué de 145 fichiers qui totalisent 183 248 mots.

Le corpus de référence a été séparé en deux : une partie destinée à apprendre la machine à états et une partie destinée à évaluer la qualité des prédictions produites. La première est appelée ‹corpus d'apprentissage› et la seconde ‹corpus d'évaluation›.

La séparation des données a été réalisée de sorte à représenter la variété des données du corpus. Pour cela, chaque fichier constituant le corpus de référence a été divisé en deux parties : apprentissage et évaluation, et les parties d'apprentissage ont été regroupées au sein du corpus d'apprentissage (143 901 mots), de même pour les parties d'évaluation (39 347 mots). De cette manière, les corpus d'apprentissage et d'évaluation représentent la variété du corpus de référence.

3.2. Métadonnées

Les fichiers constituant le CÉFC ne sont pas homogènes, ils correspondent à des conditions de production très différentes qui conduisent à des différences importantes, aussi bien lexicales que syntaxiques. On peut alors se demander s'il est préférable de réaliser l'apprentissage des outils sur les fichiers d'une même famille ou s'il vaut mieux apprendre un seul outil sur l'ensemble des données (Béchet, Nasr & Favre 2014).

Cette variété est reflétée par l'annotation des fichiers à l'aide de métadonnées qui décrivent les conditions de production. Cette annotation est réalisée sous la forme de traits [attribut-valeur] associés à chaque fichier. Sept attributs ont été définis : *modalité*, *nbLocuteurs*, *thème*, *milieu*, *type*, *canal* et *secteur*, qui peuvent avoir une à six valeurs (v. Benzitoun & Etienne 2020 dans ce volume). Les métadonnées offrent la possibilité de partitionner l'ensemble des fichiers en parties plus homogènes. Il est ainsi possible de regrouper tous les fichiers partageant la

valeur de l'attribut *thème*, par exemple, ou la valeur de deux attributs, ou trois, et ainsi de suite. Ce type de méthode se heurte au nombre de sous-corpus qu'il est possible de constituer. Selon le nombre d'attributs partagés par les sous-corpus, on obtient un nombre de sous-corpus différents et des sous-corpus de tailles différentes et il n'est pas aisé de savoir quel est le bon niveau de regroupement.

Une manière plus souple de procéder consiste à laisser ce choix à l'ordinateur. On peut, en effet, indiquer spécifiquement pour chaque mot de chaque corpus les métadonnées de ce dernier et fournir ces traits en entrée aux différents classifieurs. C'est à la charge du classifieur d'identifier l'apport que peut constituer cette information et, le cas échéant, de l'intégrer dans ses paramètres.

Quatre traits ont été pris en compte : *nbLocuteurs*, *milieu*, *type* et *secteur*. Les autres ont été omis car ils ne présentaient pas une distribution intéressante. Le trait *modalité*, par exemple, possède la même valeur pour tous les fichiers. Le Tableau 1 présente les différentes valeurs pour ces quatre traits.

Tableau 1 : Valeurs des quatre traits conservées par le classifieur et représentant les métadonnées sur les fichiers ORFÉO

nbLoc	milieu	type	secteur
1	amical, assistance	consultation, conversation	privé
2	associatif, familial	discours, entretien	professionnel
2+	médical, n/a	finalisé, narration	
	politique, scolaire	repas, réunion	

3.3. Résultats

La machine à états a été entraînée sur les données d'apprentissage et les prédictions produites sur les données d'évaluation.

Les prédictions ont ensuite été comparées aux annotations de référence. Chaque type d'annotation a été évalué à l'aide d'une mesure. Dans le cas de l'étiquetage en parties du discours et de la lemmatisation, la mesure utilisée est l'‹exactitude› *(accuracy)* qui est le rapport du nombre de prédictions correctes sur le nombre total de prédictions, ce qui donne pour le premier le PAS *(Part of speech Accuracy Score)* et pour le second le MAS *(Lemmatisation Accuracy Score)*. L'évaluation de l'analyse syntaxique a été réalisée à l'aide des deux mesures UAS *(Unlabeled Accuracy Score)* et LAS *(Labeled Accuracy Score)*. La première mesure le nombre de mots dont le gouverneur correct a été identifié. La seconde ajoute à cela la nature de la relation syntaxique liant gouverneur et dépendant. La segmentation en énoncé a été évaluée à l'aide de la précision SPS *(Sentence Precision Score)*, du rappel SRS *(Sentence Recall Score)* et de leur moyenne harmonique SFM *(Sentence F-Measure)*.

Trois machines ont été évaluées. La première, appelée ‹base›, correspond à une configuration généralement utilisée pour des textes écrits, d'un style homogène. La seconde (‹audio›) ajoute à ‹base› l'information de la durée du silence entre deux mots et du changement de locuteur. La troisième (‹audio+méta›) ajoute à ‹audio› les métadonnées. Ces trois machines diffèrent les unes des autres uniquement par les modèles de traits qu'elles utilisent. Les résultats obtenus par les trois machines sont représentés dans le Tableau 2.

Tableau 2 : Évaluation des prédictions réalisées par trois machines

machine	PAS	MAS	UAS	LAS	SPS	SRS	SFM
‹base›	98.18	95.66	78.82	76.53	67.37	65.34	66.34
‹audio›	98.19	**95.74**	81.11	78.63	76.08	**80.94**	78.44
‹audio+méta›	**98.31**	95.65	**81.40**	**79.01**	**81.81**	76.94	**79.30**

La première conclusion importante que l'on peut tirer du Tableau 2 est l'apport des traits correspondants aux silences et aux changements de locuteur pour la segmentation en énoncés, qui passe de 66.34 à 78.44. Comme on pouvait s'en douter, l'ajout de ces informations est crucial pour cette segmentation. La seconde concerne l'ajout des métadonnées, qui permettent d'améliorer légèrement la qualité de l'analyse syntaxique et de la segmentation en énoncés. Plus de détails sur ces résultats sont présentés dans la section suivante.

3.4. Analyse des résultats

Les résultats présentés dans le Tableau 2 sont des moyennes calculées sur l'ensemble des fichiers de test, néanmoins ces résultats peuvent varier significativement d'un fichier à l'autre, comme le montre la Figure 3 :

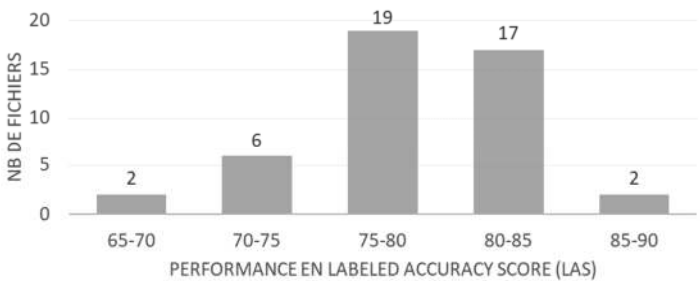

Figure 3 : Distribution des fichiers de test selon la performance moyenne de la machine ‹audio+meta›

Les performances diffèrent aussi en fonction du type de dépendance. Le Tableau 3 présente les performances en LAS en fonction des dépendances et des machines considérées. On observe que, si certaines dépendances sont presque parfaitement prédites (*spe* ou *subj*), d'autres présentent des performances bien

moindres telles que *disflink*, *para* ou *periph*. Cela peut s'expliquer pour les dépendances représentant les disfluences *(disflink)* car elles correspondent à des énoncés pour lesquelles aucune analyse satisfaisante n'est possible (les marqueurs de discours et les répétitions qui peuvent être considérés comme des liens *para* ne sont pas considérés ici). Dans ces conditions, l'analyseur a du mal à apprendre les régularités qui lui permettront de produire l'analyse correcte.

Les scores faibles des dépendances *para* et *periph* illustrent la complexité de ces relations qui relient souvent des éléments éloignés au sein des énoncés, un facteur de complexité important pour l'analyseur. De plus, la fréquence de ces relations est assez faible dans le corpus (autour de 5 % des relations) ce qui nuit à leur bonne prédiction. Enfin, les performances mitigées pour déterminer la racine des énoncés *(root)* confirment la difficulté de la tâche de segmentation des énoncés oraux.

Il est également intéressant de constater que, si l'ajout des traits ‹audio› améliore dans quasiment tous les cas les performances en LAS, ce n'est pas le cas des métadonnées, ce qui peut s'expliquer par les différences en termes de quantité des données selon les traits considérés.

Tableau 3 : Évaluation par type de dépendance (LAS) pour les trois machines et importance de chaque dépendance dans le test (en %)

dep	*aux*	*dep*	*disf*	*dm*	*mark*	*para*	*peri*	*root*	*spe*	*subj*
% test	1.9	42.2	1.2	8.0	2.8	4.4	5.2	13.6	7.8	12.6
‹base›	92.5	85.2	39.1	63.8	73.2	45.1	47.2	57.9	95.3	94.2
+‹audio›	94.2	84.9	39.5	71.4	74.9	50.7	46.7	67.2	96.3	94.1
+‹audio+méta›	93.4	86.3	39.3	75.1	74.7	47.1	49.6	62.8	96.5	94.8

Il est intéressant d'essayer de caractériser, au-delà de chaque fichier et de chaque relation, quelles sont les conditions qui affectent le plus les performances de l'analyseur syntaxique, à l'aide des métadonnées. Une telle caractérisation nous permettrait d'estimer la qualité des annotations syntaxiques automatiques produites par nos outils sur les fichiers ORFÉO qui ne font pas partie du corpus de référence annoté manuellement mais pour lesquels nous disposons de métadonnées.

Tableau 4 : Évaluation (LAS) en fonction du trait *milieu* pour les trois machines et importance de chaque trait dans le test (en %)

milieu	*amic.*	*assist.*	*asso.*	*familial*	*méd.*	*n/a*	*polit.*	*scol.*
% test	60.0	11.9	7.4	2.9	0.7	15.4	1.4	0.4
‹base›	77.4	74.6	78.2	75.6	76.2	76.7	84.6	66.3
‹audio›	79.2	79.2	80.8	78.5	76.5	77.9	84.6	68.7
‹audio+méta›	79.1	80.7	81.4	79.6	75.4	79.1	83.1	66.9

Le Tableau 4 présente les résultats en LAS en fonction des valeurs possibles du trait *milieu*. Il n'est pas évident de tirer des conclusions sur la difficulté du traitement automatique en fonction du milieu à partir de ces résultats car les scores les plus faibles (*scolaire* et *médical*) correspondent aux conditions les moins représentées dans le corpus. Il est donc difficile de déterminer si la faiblesse des performances provient d'un manque de données d'apprentissage ou bien d'une difficulté intrinsèque à ces valeurs du trait *milieu*.

Il est toutefois intéressant de constater que les résultats les plus élevés sont obtenus sur la condition *politique* bien qu'elle ne représente que 1.4 % des données. Cela est sûrement dû au contenu plus contrôlé de cette forme d'expression.

Tableau 5 : Évaluation (LAS) en fonction du trait *type* pour les trois machines et importance de chaque trait dans le test (en %)

type	*consul.*	*conv.*	*disc.*	*entr.*	*final.*	*narr.*	*repas*	*réun.*
% test	0.7	20.0	1.4	42.0	11.9	0.4	4.9	18.8
‹base›	76.2	74.0	84.6	78.5	74.6	85.8	79.5	77.1
‹audio›	76.5	76.7	84.6	79.8	79.2	90.8	83.0	78.6
‹audio+méta›	75.4	77.1	83.1	79.7	80.7	90.1	82.8	79.5

Le Tableau 5 est plus informatif dans la mesure où la corrélation entre la quantité de données et les performances n'est pas aussi directe que dans le tableau précédent.

Comme nous pouvions le supposer, les meilleurs résultats sont obtenus sur la condition *narration* bien qu'elle ne représente que 0.4 % des données. De même, la condition *conversation* affiche des performances plus faibles que les autres conditions bien que représentant 20 % des données. Cela s'explique sans doute par la nature plus spontanée de ce type d'oral en comparaison des conditions *entretien* ou encore les conversations *finalisées* enregistrées entre des usagers et des opérateurs d'un centre d'appel.

En résumé, cette analyse montre que les métadonnées sont porteuses d'une information utile pour des tâches de TAL mais que les conditions expérimentales dans lesquelles nous nous trouvons (des quantités de données différentes pour

les différentes valeurs possibles des métadonnées) rendent la comparaison des performances des outils selon les valeurs des métadonnées, difficile.

4. CONCLUSION

Nous avons décrit, dans cet article, les outils d'annotation linguistique qui ont été développés dans le cadre du projet ORFÉO et utilisés pour annoter les données diffusées. Les traitements effectués sont classiques en TAL : segmentation en mots, étiquetage en parties de discours, lemmatisation, analyse syntaxique et segmentation en phrases. Ces différentes étapes ont été regroupées au sein d'un modèle unique, appelé ‹machine à états›. L'originalité de cette machine est qu'elle réalise un traitement incrémental de l'énoncé : un mot est traité à différents niveaux d'analyse avant de passer au mot suivant alors qu'en général les traitements sont séquentiels : l'intégralité de l'énoncé est traitée à un niveau donné avant de passer au niveau suivant. La raison principale pour laquelle nous avons opté pour cette approche est liée à la segmentation en phrases. Dans le cadre de l'écrit, qui est le cas dominant pour les applications du TAL, la segmentation en phrases est le premier traitement réalisé. Cela se justifie dans la mesure où les frontières de phrases sont généralement faciles à déterminer à l'aide de la ponctuation. Ce n'est pas le cas pour l'oral et une segmentation réalisée prématurément conduit généralement à de mauvais résultats. Dans l'approche que nous proposons, la segmentation en phrases constitue la dernière étape des traitements. Elle est réalisée après l'analyse syntaxique, ce qui permet de prendre en compte la structure syntaxique des énoncés pour identifier une fin de phrase. De plus, le modèle que nous proposons permet facilement de prendre en compte des indices issus du signal de parole (dans notre cas, les pauses et les changements de locuteurs) dans les traitements. Nous avons montré que l'utilisation conjointe d'informations syntaxiques et prosodiques permet d'améliorer nettement la qualité de la segmentation en phrases.

Le second point original dans l'architecture proposée est la prise en compte des métadonnées dans les traitements linguistiques. Les données constituant le CÉFC sont hétérogènes, se distinguant les unes des autres par le nombre de locuteurs, la nature de la conversation ou encore son thème. Cette variété induit des différences dans les choix lexicaux et grammaticaux effectués par les interlocuteurs. On peut se demander, dans ce cas, s'il est préférable de produire des modèles différents pour chaque type de conditions ou s'il vaut mieux produire un seul modèle qui reflétera la moyenne des variations observées entre les corpus. Chacun de ces deux choix possède ses faiblesses. Dans le premier cas, la séparation des données en sous-parties homogènes aboutit à des corpus de taille réduite et, par conséquent, en général à des modèles de moindre qualité. Dans le second cas, la production d'un modèle unique ne permet pas de bien modéliser certains phénomènes qui sont spécifiques à des sous-parties du corpus. La solution que nous proposons à ce problème est d'expliciter, à l'aide des métadonnées d'ORFÉO, les spécificités de chaque sous-partie et de laisser les

outils informatiques identifier les spécificités qu'il est utile de prendre en compte dans les traitements.

Références

[ORFÉO] *Corpus d'Étude pour le Français Contemporain*, ATILF, LIF, Loria, CLLE-ERSS, ICAR, LaTTiCe. [https://www.ortolang.fr/market/corpora/cefc-orfeo]

[TCOF] *Traitement de Corpus Oraux en Français*, ATILF (CNRS & Université de Lorraine). [http://www.cnrtl.fr/corpus/tcof/]

BÉCHET F., NASR A. & FAVRE B. (2014), "Adapting dependency parsing to spontaneous speech for open domain spoken language understanding", *Proceedings of the 15th Annual Conference of the International Speech Communication Association – Interspeech 2014* (Singapore), 135-139.

BENZITOUN C. & ETIENNE C. (2020), « Méthodologie d'harmonisation et de traitement des données orales du CÉFC », *Langages* 219. (ce volume)

CERISARA C., MELLA O. & FOHR D. (2009), "Jtrans: An open-source software for semi-automatic text-to-speech alignment", *Proceedings of the 10th Annual Conference of the International Speech Communication Association – Interspeech 2009* (Brighton, UK), Baixas, International Speech Communications Association, 1799-1802.

DEULOFEU J. & VALLI A. (2020), « Lexique et classement en parties du discours dans ORFÉO », *Langages* 219. (ce volume)

MIKOLOV T. *et alii* (2013), "Distributed representations of words and phrases and their compositionality", in C.J.C. Burges *et alii* (eds.), *Advances in Neural Information Processing Systems* 26, San Diego (CA), Neural Information Processing Systems Foundation, Inc., 3111-3119.

NASR A. *et alii* (2011), "Macaon: An NLP tool suite for processing word lattices", *49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies – ACL-HLT 2011*, *Proceedings of System Demonstrations* (Portland, Oregon, USA), Stroudsburg (PA), Association for Computational Linguistics, 86-91.

NASR A. *et alii* (2015), "Joint dependency parsing and multiword expression tokenization", in C. Zong & M. Strube (eds.), *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics* (Beijing, China), vol. 1, Stroudsburg (PA), Association for Computational Linguistics, 1116-1126.

NIVRE J. *et alii* (2007), "The CoNLL 2007 shared task on dependency parsing", in J. Eisner (ed.), *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning – EMNLP-CoNLL* (Prague, Czech Republic), Stroudsburg (PA), Association for Computational Linguistics, 915-932.

NIVRE J. *et alii* (2016), "Universal dependencies v1: A multilingual treebank collection", in N. Calzolari *et alii* (eds.), *Proceedings of the Tenth International Conference on Language Resources and Evaluation – LREC'16* (Portorož, Slovenia), Paris, European Language Resources Association (ELRA), 1659-1666.

RAMISCH C. *et alii* (2016), "DeQue: A lexicon of complex prepositions and conjunctions in French", in N. Calzolari *et alii* (eds.), *Proceedings of the Tenth International Conference on Language Resources and Evaluation – LREC'16* (Portorož, Slovenia), Paris, European Language Resources Association (ELRA), 2293-2298.

RUMELHART D. E., HINTON G. E. & WILLIAMS R. J. (1986), "Learning representations by back-propagating errors", *Nature* 323, 533-536. ["Learning internal representations by error propagating", *ICS Report 8506*, 1985, La Jolla (CA), Institut for Cognitive Science, 34 p.]

YAMADA H. & MATSUMOTO Y. (2003), "Statistical dependency analysis with support vector machines", *Proceedings of the Eighth International Conference on Parsing Technologies – IWPT 2003* (Nancy, France), 195-206. [en ligne]