Détection automatique d'erreurs d'annotations pour améliorer les performances des algorithmes d'apprentissage automatique

Carole Lemort

Rapport de stage

Master 2 Recherche en Informatique

Détection automatique d'erreurs d'annotations pour améliorer les performances des algorithmes d'apprentissage automatique

Carole LEMORT

Maître de stage : Christian Raymond

Équipe : TEXMEX

Mars - Juillet 2011

Remerciements

Tout d'abord, je tiens à remercier Patrick et Christian pour leur soutien et leurs conseils.

Ensuite, je tiens aussi à remercier tous les membres de l'équipe pour leur accueil et leur sympathie. En particulier, Florent pour nos grandes discussions sur le traitement automatique des langues et le langage Perl.

Enfin, je voudrais remercier tout le personnel de l'INRIA et spécialement mes collègues de l'équipe BUNRAKU.

Merci aussi à Mathilde et à Jonathan.

Résumé

Actuellement, les techniques de transcriptions automatiques de parole et d'annotation de ces transcriptions restent très dépendantes de l'environnement dans lequel ont lieu les enregistrements. Cela pose un problème pour le résumé automatique, la traduction, ainsi que la reconnaissance d'entités nommées. Aussi le but de ce document est d'améliorer la détection des annotations erronées afin qu'il soit possible par la suite les corriger et d'obtenir des annotations de meilleure qualité.

Dans un premier temps, nous présentons un état de l'art, puis le protocole expérimental. Dans un second temps, nous commentons les résultats.

Mots clefs : Traitement automatique du langage naturel, apprentissage automatique, annotations, détection.

Table des matières

Liste des tableaux

Introduction

Avec l'évolution de la technologie, le stockage et le traitement des fichiers de données de toutes sortes est en plein essor.

À l'heure actuelle, tous les ordinateurs, appareils photos, téléphones portables, ... sont capables d'enregistrer des données de différents types (textes, sons, vidéos, images, ...). Parmi ces masses de données, la plupart sont enregistrées puis stockées sans jamais être réutilisées, faute de temps ou d'outils efficaces. Or si nous voulons que ces données puissent être utiles, il est nécessaire de les classer mais aussi de pouvoir les utiliser et réutiliser facilement.

Dans le cadre du traitement automatique des langues, et en particulier de la parole, les données que nous voulons analyser sont les termes prononcés par des humains. Par exemple, cela peut être des transcriptions d'émissions radiophoniques ou télévisuelles, une conversation téléphonique, ou simplement un texte dicté dans un microphone. L'analyse de transcriptions possède un grand nombre d'applications parmi lesquelles la traduction automatique et la recherche d'informations.

Le traitement automatique des langues est un domaine qui nécessite des connaissances en linguistique, en informatique et en intelligence artificielle dans lequel de nombreux types de modèles sont utilisés. Néanmoins, dans le domaine du traitement automatique de la parole, les modèles formels sont inefficaces car ils sont trop rigides. En effet, dans la langue orale, la grammaire est approximative. De plus, des reprises et des hésitations sont inévitables. D'ailleurs, les transcriptions de dialogues ne comportent ni ponctuation, ni majuscules. En plus de cela, les transcriptions automatiques sont imparfaites étant donné que les mots ne sont pas toujours bien reconnus. C'est pourquoi les méthodes utilisées sont des méthodes statistiques.

L'inconvénient des modèles statistiques est qu'ils ont besoin de corpus annotés pour être entraînés. Ces corpus annotés coûtent chers et peuvent contenir des erreurs. En effet, les corpus sont annotés par un grand nombre de personnes. De plus, chacun peut avoir sa propre interprétation des consignes d'annotation et des phrases à annoter.

D'autre part, les incohérences d'annotation ont des répercussions sur les performances des algorithmes. C'est pourquoi il est crucial que les données soient les plus exactes possible, en particulier dans le cas des algorithmes d'apprentissage automatique. Par conséquent, les erreurs qui sont sources d'inexactitude, doivent pouvoir être détectées afin d'être soit retirées, soit corrigées.

Afin que les annotations des transcriptions soient les meilleures possible, nous allons avoir pour but de repérer les annotations incorrectes.

Nous commencerons d'abord par présenter un état de l'art. Ensuite, nous présenterons le protocole expérimental. Enfin nous verrons les premiers résultats et pourrons conclure sur le travail déjà effectué.

Chapitre 1

État de l'art

Cette première partie a pour objectif de faire un état de l'art sur les travaux existants en correction automatique d'annotation. Pour cela, nous nous intéresserons d'abord aux données pour lesquelles on cherche à améliorer les annotations. Nous verrons ensuite les différents algorithmes traditionnellement utilisés pour générer ces annotations. Et enfin, nous verrons le peu de techniques existantes pour corriger automatiquement ces annotations.

1.1 Données utilisées

Les données que l'on utilise sont issues de corpus de données annotés. Les énoncés extraits de ces ensembles de données sont interprétées comme des phrases, donc des séquences des termes avec un ordre et non pas simplement comme des sacs de mots. Une phrase contenant généralement plusieurs mots, en général elle possède aussi plusieurs annotations. Comme nous nous intéressons à des problèmes d'annotation de séquences, il est important de savoir que dans une phrase les annotations ne sont pas toutes vraies ou toutes fausses. Pour éviter de perdre les informations exactes contenues dans les annotations correctes d'une phrase qualifiée d'erronée, le but est de tenter de repérer et corriger les annotations incorrectes.

1.1.1 ATIS

Le corpus Air Travel Information System (ATIS) est utilisé pour fournir des informations sur les voyages aériens [5]. À l'origine, la base de données ne contenait les informations des aéroports que de 11 villes, mais la tâche d'indexation d'informations était limitée uniquement à cause de la petite taille de la base de données. Aussi le nombre d'informations disponibles pour l'étude a été fortement augmenté, la nouvelle base de données contient les informations de 52 aéroports situés dans 46 villes des États-Unis et du Canada. Finalement, la plus grande table de la base inclut les informations de 23 457 vols. Le passage à une base de données trois fois plus grande n'a pas posé de difficultés particulières, ce qui est encourageant car cela montre que pour ce type de système, une petite base de données peut être mise à l'échelle pour une tâche plus importante.

TAB. 1.1 – Exemple d'annotation conceptuelle sur un énoncé extrait du corpus ATIS

Numéros	Mots	Étiquette
1	information	null
2	on	null
3	american-airline	airline_name
4	flights	null
5	from	null
6	washington	fromloc.city_name
7	to	null
8	philadelphia	toloc.city_name
9	early-morning	depart_time.period_of_day
10	times-of-flight	flight_time

Dans le corpus ATIS, au fur et à mesure de l'avancement, les processus d'annotation ont été automatisés. En effet, le corpus original a été enregistré selon un protocole de *Magicien d'Oz*. Dans un protocole de *Magicien d'Oz*, les utilisateurs croient qu'ils sont en relation avec un ordinateur, alors qu'en réalité ils discutent avec un humain (le magicien), qui simule le comportement d'un serveur vocal d'informations. Ensuite, le nombre de dialogues collectés est devenu de plus en plus important. Les transcriptions et les interprétations des demandes émises par les utilisateurs ont alors été réalisées de manière automatique. Le traitement automatique des données ayant assurément l'avantage de réduire le coût de la procédure. En revanche, cela provoque l'existence d'artéfacts dans la collection, tel que le même énoncé répété un grand nombre de fois.

Pour chaque session utilisateur, nous disposons des éléments suivants :
- le son de la requête en audio numérique
- l'historique d'événements de la session
- la transcription détaillée
- la réponse de référence minimale
- la réponse de référence maximale
- la catégorie de la requête

Les données récupérées sont classées en 3 catégories : indépendant du contexte pour obtenir l'interprétation, dépendant du contexte pour produire l'interprétation et non évaluable.

Au début, le corpus contenait 12 047 dialogues dont 3 876 avaient été annotés, environ le tiers des dialogues étaient correctement annotés par rapport aux informations contenues dans la base de données comme nous pouvons le voir dans l'exemple l'exemple 1.1 page 9.

Enfin, l'interêt des tâches téléphoniques est à la fois d'encourager la recherche sur le dialogue homme-machine mais aussi les communications téléphoniques.

1.1.2 MEDIA

La campagne d'évaluation MEDIA fait partie du programme français Technolangue Evalda. Son but est de définir et tester une méthodologie pour évaluer les capacités d'in-

terprétations de plusieurs systèmes sur un corpus de dialogue homme-machine, portant sur un serveur d'informations touristiques. L'objectif est une évaluation qui s'effectue de manière automatique, et qui permet de comparer différents systèmes entre eux sur des critères similaires, [6] et [7].

Afin d'obtenir le corpus, 250 locuteurs ont effectué chacun 5 scénarios de réservation d'hôtels avec un système de dialogue simulé par un opérateur humain. 1257 dialogues d'une durée moyenne de 3 minutes 30 ont ainsi pu être enregistrés. Au final, le corpus comporte plus de 70 heures de dialogues qui ont été transcrits manuellement puis annotés de façon décrite ci-après.

Le corpus MEDIA a été enregistré selon un protocole de *Magicien d'Oz* qui simule un serveur vocal d'informations qui donne accès à des informations touristiques, et permet la réservation de chambres d'hôtels.

La campagne comporte deux phases d'évaluation : une compréhension hors contexte et une compréhension en contexte. Hors contexte, les dialogues sont traités indépendamment les uns des autres. On cherche la représentation littérale de la signification de l'énoncé étudié de manière isolée, donc sans aucune information sur le dialogue en cours. Pour l'évaluation en contexte, on recherche la représentation sémantique du dialogue, les concepts sont alors enrichis avec les informations contextuelles obtenues grâce à l'analyse du dialogue en cours, comme dans l'exemple 1.2 page 11.

Pour permettre le décodage conceptuel, il est nécessaire de définir une représentation sémantique. Cette représentation se base sur la structure attribut-valeur qui permet un processus simple d'annotations. De plus, la hiérarchie conceptuelle permet d'identifier les relations entre les unités sémantiques. Dans le corpus MEDIA, une unité sémantique est représentée par un 5-uplet qui indique :

- le mode (positif, affirmatif, interrogatif ou optionnel)
- le nom de l'attribut représentant le sens de la séquence de mots
- la valeur de l'attribut
- les liens : pointeurs optionnels qui relient l'unité sémantique une unité précédente (uniquement utile pour la représentation sémantique contextuelle)
- un commentaire optionnel sur l'unité sémantique

Les attributs de base sont divisés en plusieurs classes. Les attributs de la base de données sont classées en paquets qui peuvent être indépendants du domaine, tels que des unités numériques ou des dates, ou dépendants du domaine tels que des noms d'hôtels. Par ailleurs, comme on peut le voir dans le tableau 1.2, chaque paquet est défini comme une hiérarchie d'attribut, par exemple, le paquet *paiement* implique un sous-attribut *montant* qui lui-même implique un sous-attribut *nombre*. Dans l'exemple précédent, on constate que la valeur *cent dix* occupe ce rôle.

Les attributs modifieurs sont quant à eux utilisés pour modifier le sens des attributs de la base de données. Ils permettent de compléter l'interprétation des prix ou des distances. Ainsi l'attribut comparatif ayant pour valeur *inférieur*, ici associé à un montant monétaire : *cent dix euros*, nous renseigne sur le type de recherche automatique à effectuer dans la base de données hôtelière.

TAB. 1.2 – Exemple de représentation sémantique, extrait de [6]

Numéro	Mots	Mode	Nom de l'attribut	Valeur de l'attribut
0	euh	+	null	
1	oui	+	réponse	oui
2	l'	+	lien	singulier
3	hôtel	+	objet de la base de données	hôtel
4	dont	+	null	
5	le prix	+	objet	paiment-montant-chambre
6	ne dépasse pas	+	*paiement*-comparatif	inférieur
7	cent dix	+	*paiement-montant-nombre*-chambre	110
8	euros	+	*paiement*-devise	euro

Les attributs généraux définissent des actions de commande du dialogue tels que annulation, correction, demande d'informations ...

Finalement, la représentation comporte 73 attributs de la base de données, 4 modifieurs et 6 attributs généraux, ce qui forment au final 83 attributs.

Malgré l'utilisation de la représentation attribut-valeur qui a pour but de simplifier et d'uniformiser le travail des annotateurs humains, la tâche d'annotation reste difficile. En effet, les annotations sont sensibles à l'interprétation des annotateurs et aux erreurs humaines. Aussi pour l'évaluation du corpus MEDIA, un coefficient kappa a été mis en place pour vérifier la qualité des annotations. Dans le cas présent, le coefficient kappa k est calculé au niveau des attributs à l'aide des deux statistiques suivantes : $P(A)$ qui est la proportion du temps où les annotateurs sont d'accord, et $P(E)$ qui est la probabilité qu'un annotateur choisisse le concept correct par hasard. Le coefficient kappa qui normalise la complexité de la tâche d'annotation vaut $\frac{P(A)-P(E)}{1-P(E)}$.

Dans la littérature, un coefficient kappa supérieur à 0.8 est généralement considéré comme bon. Pour le corpus MEDIA, cela correspond à un accord inter-annotateur de plus de 80%, il aura fallu trois étapes aux annotateurs pour atteindre ce score [6]. De plus, après quelques passages sur les données, l'accord inter-annotateur est systématiquement supérieur à 80% (kappa supérieur à 0.8), et dans le meilleur des cas, l'accord inter-annotateur atteint même presque 90%. Lorsque ce corpus a été réalisé, ce chiffre était assez bon pour distribuer les données aux participants de la campagne qui ont du adapter leur compréhension du modèle à la tâche demandée. Malgré tout, obtenir un accord inter-annotateur parfait reste difficile, le consortium MEDIA prévoyait donc une double annotation pour la partie test du corpus.

1.1.3 ESTER

La campagne Évaluation des Systèmes de Transcriptions enrichie d'Émissions Radiophoniques, présentée dans [8] et [9] a pour but l'évaluation objective des performances, dans le domaine du traitement automatique de la parole et du langage naturel. Il existe

peu de corpus en langue française, l'objectif étant ici l'évaluation et le développement de corpus et de protocoles. Pour cela, il est nécessaire d'établir des métriques d'évaluation simples et homogènes pour permettre la comparaison des résultats.

Le corpus est composé de 100 heures d'émissions radiophoniques francophones issues des stations France Inter, France Info, Radio France International, France Culture, Radio Classique et Radio Télévision Marocaine émises en langue française. Les 100 heures d'émissions se décomposent de la manière suivante : 82 heures d'apprentissage, 8 heures de développement et 10 heures de test.

Le fait d'utiliser des émissions radiophoniques présente deux intérêts principaux. En premier lieu, si l'on compare ce corpus avec ceux déjà existants en langue française, on constate un accroissement de la difficulté en raison de la taille plus importante du vocabulaire utilisé. Et de plus, les tâches choisies pour être réalisées dans le cadre de cette campagne ont un potentiel intéressant en termes d'applications pratiques dans le domaine du traitement des langues.

Les tâches réalisées durant la campagne ESTER se décomposent en 3 catégories : transcription, segmentation et extraction d'informations. La transcription consiste en la transcription orthographique et la transcription temps-réel. La segmentation comporte 4 opérations qui consistent à suivre des événements ou des locuteurs. L'extraction d'informations comprend elle aussi 4 traitements qui ont pour but de permettre l'enrichissement des transcriptions avec des informations de plus haut niveau. Parmi les différentes tâches réalisées durant la campagne ESTER, de notre point de vue, la plus intéressante ici est la détection d'entités nommées. La détection d'entités nommés consiste à détecter dans les documents audios les occurrences d'entités identifiées. Ces entités nommées apportent des informations qui sont utiles pour classer les transcriptions et permettre la recherche d'informations. De plus, les entités nommées sont utilisées dans de nombreux processus du traitement des langues tels que la traduction ou le résumé automatique.

Dans le cas présent, la classification des entités nommées dépend des conventions utilisées selon la campagne ESTER utilisée pour les expérimentations. Lors de la première campagne, 30 types d'entités nommées furent utilisées. Ces entités sont classées en 9 catégories principales : personne, organisation, groupe géo-socio politique, lieu, bâtiment et construction humaine, production humaine, date et heure, montant et inconnue. Pour la deuxième campagne, 37 types d'entités nommées furent réparties en 7 catégories : personne, fonction, organisation, lieu, production humaine, date et heure, et montant.

Afin de pouvoir apporter de l'information supplémentaire à la transcription, il existe des listes contenant des noms de lieux, de personnes ou de fonctions importantes. Ceux sont les occurrences de ces entités nommées que l'on recherche afin de savoir si le dialogue en cours d'étude possède un intérêt particulier.

1.2 Algorithmes d'apprentissage automatique pour la classification de séquences

La classification de séquences est un problème d'étiquetage de séquences dont le but est d'associer à une séquence donnée, la séquence d'étiquettes correspondante la plus probable. Dans le traitement automatique de la parole, la séquence donnée est une suite de mots et la séquence d'étiquettes recherchée est la séquence de concepts correspondante. Dans le traitement de la parole, la suite de mots est une séquence de vocables issus de la transcription de la parole. Pour résoudre le double problème de la segmentation et de l'étiquetage (trouver le concept ainsi que ses frontières dans la séquence de mots), l'encodage BIO qui permet la structuration du texte est généralement utilisé. Un indicateur B (pour begin), I (pour inside) ou O (pour outside) est ajouté à l'étiquette pour permettre de savoir si le mot correspondant est au début, à l'intérieur ou à l'extérieur du concept comme on peut le voir dans l'exemple suivant :

TAB. 1.3 – Exemple d'encodage BIO sur des données du corpus ESTER, extrait de [1]

Position	-3	-2	-1	0	+1	+2	+3
Mots	ici	jacques	doutisoro	lomé	africa	numéro	un
Attributs	ici	NPMS	inconnu	VILLE	NPSIG	numéro	un
Étiquettes	O	pers-B	pers-I	loc-B	org-B	org-I	org-I
Entités	O	pers		loc	org		

Par exemple, les auteurs de [1] utilisent différents niveaux de description au niveau des attributs afin de permettre un décodage conceptuel plus précis. Le premier niveau de description utilisé est simplement les mots de la transcription. Ensuite le résultat d'un étiquetage morpho-syntaxique peut être utilisé, par exemple, NPMS qui indique un nom propre masculin singulier. Le second niveau de description équivaut quant à lui à un mélange de différents éléments tel que l'étiquetage précédent et des connaissances connues *a priori* telles que des listes de pays, villes ou unités de mesure. Et le troisième niveau correspond à des mots considérés comme important en fonction du corpus étudié, c'est à dire des mots à capacité de généralisation satisfaisante. Dans l'exemple, l'étiquette courante est estimée à partir des mots et des attributs dans l'entourage immédiat (-1, +1 ou -2, +2) autour du mot courant pour lequel on veut prendre un décision.

Nous allons maintenant présenter différents types d'algorithmes traditionnellement utilisés en étiquetage de séquences.

1.2.1 Classifieurs locaux

Lorsque l'on cherche à étiqueter une phrase, l'ensemble des mots de cette phrase est vu comme une séquence, mais le problème peut aussi être envisagé comme une suite de classifications locales, et il est possible de classifier chaque terme de la séquence indépendamment, à l'aide d'algorithmes de classification locale, par exemple, les machines à vecteurs de support ou le boosting.

Le but du boosting est d'améliorer la précision des règles de classification en combinant plusieurs hypothèses dites faibles [12]. Les algorithmes de boosting travaille en

re-pondérant répétitivement les exemples mal classés dans le jeu d'entraînement, et en ré-éxécutant l'algorithme d'apprentissage faible sur ces données re-pondérées. Le système d'apprentissage se concentre ainsi sur les exemples les plus compliqués. Les hypothèses faibles sont ensuite combinées par un vote pondéré pour créer la règle de classification finale. Le boosting permet d'associer un ou plusieurs étiquettes à l'énoncé étudié.

Afin de classer les dialogues étudiés de manière automatique, plusieurs méthodes d'apprentissage sont utilisées, l'une des premières à avoir donné des résultats en décodage conceptuel a été introduite par Vapnik, ce sont les machines à vecteurs de support. Les machines à vecteurs de support (abrégées SVM : *Support Vector Machine* par la suite) sont des classifieurs discriminants à vaste marge. Les SVM sont des classifieurs binaires qui représentent les échantillons à classer sous la forme d'un vecteur dont chaque composante représente la contribution d'un paramètre à un exemple.

Lorsque l'on utilise les SVM sur une transcription, l'ordre des mots dans la phrase n'est habituellement pas considéré (technique du sac de mots). La séquence est vue comme un ensemble de mots non structuré, l'ordre des vocables est la plupart du temps perdu [12]. Malgré tout, l'utilisation des SVM est justifiée car cette technique peut gérer de grandes quantités de données, et les vecteurs représentant les différentes phrases du dialogue contiennent peu de données non nulles. Or les SVM sont adaptés aux problèmes ayant des vecteurs creux.

Dans ce type de méthode, il est important de remarquer que chaque étiquette de la séquence est traitée indépendamment des autres. Durant l'apprentissage, il existe deux méthodes pour étendre les classifieurs locaux qui sont une méthode de classification binaire, donc à l'origine non multi-classes, aux nombres de labels voulus. Soit on détermine N classifieurs binaires et un vote est effectuée pour déterminer la classe. C'est la méthode *one-versus-all*. Soit on construit $M(M-1)/2$ classifieurs binaires en opposant les différentes classes. Cette méthode dite du *one-versus-one* fait passer l'échantillon à analyser dans chaque classifieur et un vote permet de déterminer la classe de l'échantillon.

1.2.2 Modèles de Markov

L'algorithme à base de transducteurs à états fini est une approche générative stochastique. Cette approche se base sur le calcul de la probabilité jointe entre la séquence d'observations et la séquence d'étiquettes. Les transducteurs à états fini (abrégés FSM : *Finite State Machines*) sont des modèles de Markov cachés adaptés pour traiter les graphes de mots, ils permettent d'intégrer les connaissances sémantiques du système de dialogue au modèle statistique de reconnaissance de la parole [3]. Pour obtenir la séquence de concepts contenus dans le dialogue étudié, l'objectif est de trouver la séquence d'étiquettes (concepts) maximisant la probabilité *a posteriori*. La probabilité *a posteriori* est calculée à l'aide de l'équation suivante :

$$P(W,C) = \prod_{i=1}^{k} P(w_i c_i | h_i) \tag{1.1}$$
$$\text{avec } h_i = \{w_{i-1}c_{i-1}, \ldots, w_1 c_1\}$$

Dans cette formule, $W = w_1, w_2, \ldots, w_k$ est la séquence de mots, et $C = c_1, c_2, \ldots, c_k$ la séquence de concepts. h_i représente quant à lui l'historique, il dépend du modèle n-gramme utilisé.

Afin de déterminer la séquence de concepts présents dans la transcription, une grammaire régulière est définie pour chacun de ces concepts. De plus, pour éviter des traitements non appropriés, les transducteurs peuvent repérer les séquences de mots où aucune notion pertinente n'est présente. Ces modèles de Markov cachés adaptés au traitement du langage peuvent associer plusieurs interprétations pour une même séquence de mots mais une seule segmentation d'une chaîne de mots est possible pour une séquence donnée de concepts. Au final, on obtient une chaîne de mots si seul les symboles d'entrées sont pris en compte et une séquence d'étiquettes conceptuelles si on garde uniquement les symboles de sortie.

Ensuite, on recherche la liste des meilleures interprétations. L'avantage d'utiliser une liste structurée des n meilleures interprétations possibles est que chaque hypothèse a un sens différent pour l'utilisation du dialogue.

1.2.3 Champs conditionnels aléatoires

A l'inverse des modèles génératifs qui ont besoin d'énumérer toutes les séquences d'observations possibles pour pouvoir définir une probabilité jointe, les classifieurs de séquences tels que les modèles de Markov d'entropie maximale ou les champs conditionnels aléatoires n'ont pas besoin d'une énumération exhaustive. De plus, les champs conditionnels aléatoires permettent qu'une étiquette dépende de l'observation courante mais aussi d'observations passées et futures contrairement aux modèles génératifs qui suppose une indépendance stricte (limitée à l'historique) entre les différents attributs. En outre, les champs conditionnels aléatoires réussissent à combiner les avantages des deux types de modèles. En effet, ils peuvent manipuler un grand type d'attributs mais aussi gérer des dépendances entre les étiquettes de sortie. Enfin, ils prennent une décision globale sur la séquence.

Un champ conditionnel aléatoire, défini dans [4], est caractérisé par un graphe de dépendances G et un ensemble de fonctions f_k auxquelles sont associées des poids λ_k. La probabilité conditionnelle d'une annotation, séquences d'étiquettes y est calculé en fonction des observations x grâce à la formule suivante :

$$p(y|x) = \frac{1}{Z(x)} \exp\left(\sum_{c \in C} \sum_{k} \lambda_k f_k(y_c, x, c)\right) \tag{1.2}$$

$$\text{avec } Z(x) = \sum_{y} \exp\left(\sum_{c \in C} \sum_{k} \lambda_k f_k(y_c, x, c)\right)$$

Les observations x utilisées pour déterminer la probabilité ci-dessus sont les mots, les étiquettes morpho-syntaxiques, et parfois les connaissances *a priori*. Les descripteurs (lexicaux, sémantiques, ...), les connaissances et les relations entre concepts sont encodés dans le modèle à travers les fonctions f_k. Ces fonctions binaires retournent 1 s'il y a correspondance ou 0 sinon. Elles prennent en paramètre les valeurs des variables aléatoires y_c de la clique c à laquelle elles s'appliquent, ainsi que l'ensemble des observations x. Les poids λ_k associés à chacune de ces fonctions sont les paramètres du modèle estimés lors de la phase d'apprentissage. Apprendre un champ conditionnel aléatoire consiste donc à calculer les poids λ_k.

Les champs conditionnels aléatoires présentent tous les avantages des modèles de Markov à entropie maximale. Ils peuvent donc être considérés comme des modèles à états fini ayant des probabilités de transition non normalisées [4]. D'ailleurs d'après les expériences menées sur les champs conditionnels aléatoires que l'on peut trouver dans la littérature, par exemple dans [2], ils obtiennent de meilleurs résultats que les modèles de Markov à états cachés et les modèles de Markov à entropie maximale, quand la distribution des données présente des dépendances plus importantes que dans le modèle, ce qui est souvent le cas dans la réalité.

1.3 Détection d'erreurs et correction automatique d'annotations

Dans le cadre du traitement automatique du langage naturel, il est nécessaire lorsque l'on recours à l'apprentissage automatique d'avoir à disposition de grands corpus d'apprentissage. Or ces corpus de données annotées coûtent chers, sont difficiles à obtenir et nécessitent beaucoup de temps.

Aussi le but de l'apprentissage automatique en traitement du langage est de traiter et d'annoter ces données de manière systématique [14]. Le problème ici est que pour l'apprentissage automatique, il est important d'avoir de grandes quantités de données de bonne qualité.

Des méthodes telles que l'apprentissage actif ont été développées pour diminuer le coût lié à l'annotation des corpus. Le but de l'apprentissage actif est de diminuer le nombre de dialogues à annoter en sélectionnant les dialogues les plus informatifs. En effet, intégrer un algorithme de détection d'erreurs dans les processus d'apprentissage actif peut être très bénéfique [10]. En premier lieu, détecter les erreurs d'annotations permet d'améliorer la qualité des données d'entraînement afin d'apprendre un modèle plus précis. Et dans

un second temps, détecter les erreurs montre quelles annotations sont incorrectes, et peut aider à augmenter les performances de l'outil d'annotation automatique à chaque passage de l'algorithme.

Afin d'augmenter les connaissances de l'algorithme, il est nécessaire que l'algorithme puisse communiquer avec les données sur lesquelles il apprend. Normalement les données sont utilisées par les algorithmes pour apprendre, mais l'algorithme peut aussi donner un retour d'informations sur les données, c'est le principe de l'apprentissage actif. L'algorithme d'apprentissage est entraîné sur un petit jeu de données annotées, puis est ensuite utilisé pour annoter automatiquement un jeu de données non annoté. Les données pour lesquelles l'algorithme est le moins confiant dans son analyse seront passées à des annotateurs humains. Ce sont les exemples pour lesquels l'algorithme ne sait a priori pas faire. Ces données vont donc fournir beaucoup d'informations une fois annotées correctement par des humains.

De plus, les annotations sont affectés par les erreurs humaines et les méthodes d'apprentissage automatique sont très sensibles à la qualité des données analysées. Comme on peut le voir dans [1] il faut éviter d'augmenter la quantité de données nécessaires à l'algorithme sans prêter attention à leur justesse, en particulier dans un contexte de transcription de parole. Effectivement, les mots à analyser étant limités au lexique défini par le système de reconnaissance, et la robustesse du système étant essentielle, il est nécessaire que les annotations soient de la meilleure qualité possible.

Par ailleurs, avant de tenter de corriger les erreurs d'annotation, intéressons nous à la détection de ces erreurs. La détection d'anomalies est une méthode qui consiste à déterminer quels éléments d'un ensemble de données ne sont pas conformes à la globalité. Les erreurs d'annotation sont des données incohérentes par rapport à une collection, elles peuvent être considérées comme des anomalies. La plupart des travaux sur la détection d'anomalies sont issus d'études statistiques sur la recherche de données aberrantes (*outliers*). Dans la méthode présentée dans [15], une distribution de probabilité est tout d'abord calculé sur sur le corpus entier. Puis un test statistique est effectué pour identifier quels sont les éléments incohérents. Grâce à cette méthode, il est possible de détecter une partie des données mal annotées.

En plus de pouvoir détecter les erreurs d'annotations qui faussent les algorithmes, on peut même réussir à corriger ces erreurs de manière automatique. Assurément, cela nécessite d'abord de détecter les annotations incohérentes afin de pouvoir les corriger ou parfois simplement les préciser. Le but étant bien évidemment ici de rendre les annotations homogènes.

Dans l'article de 2010 Reconnaissance robuste d'entités nommées sur de la parole transcrite automatiquement, paru à la conférence TALN, une expérience menée sur des données du corpus ESTER (décrit section 1.1.3 page 11) démontre que la correction automatique d'annotations est possible dans certaines conditions.

Un premier niveau de description des données est composé de mots de la transcription

et du résultat d'un étiquetage morpho-syntaxique. Un deuxième niveau comporte quant à lui des mots considérés comme importants et des classes de généralisation de connaissances connues *a priori* telles des listes de pays, de villes, d'unités de mesures, ... en plus des mots et des étiquettes morpho-syntaxiques.

Grâce à ces deux niveaux de description, on apprend tout d'abord des règles faibles, donc peu généralisables, sur un ensemble d'apprentissage à l'aide du premier niveau de représentation. Ensuite, à l'aide d'un corpus fiable décrit avec le deuxième niveau d'observation, qui lui a produit des règles plus fortes et par conséquent applicables à un plus grand nombre de données, on analyse de nouveau le corpus d'entraînement avec le nouveau niveau de description. Ce corpus d'entraînement est donc ré-annoté de manière automatique avec les nouvelles règles. Enfin, cette nouvelle annotation pourra servir de référence pour l'apprentissage afin d'optimiser les résultats.

Chapitre 2

Protocole expérimental

Dans cette seconde partie, nous verrons tout d'abord le processus d'apprentissage et d'annotation puis les différentes versions du corpus ainsi que leurs rôles. Ensuite nous développerons les calculs des résultats de référence qu'il est nécessaire d'effectuer afin d'avoir une base de référence à laquelle comparée les résultats des expériences.

2.1 Description du protocole

Pour commencer, un modèle est appris sur un ensemble de données en fonction d'un motif qui généralement prend en compte une fenêtre définie autour du mot courant.

Les différents modèles d'apprentissage utilisés pour réaliser les expériences sont des arbres de décisions sémantiques, des machines à vecteurs de support et des transducteurs à états fini (modèles de Markov cachés adaptés pour traiter des graphes de mots). Deux logiciels sont aussi employés : Wapiti qui permet d'apprendre un modèle à l'aide de champs conditionnels aléatoires et BonzaiBoost qui exploite quant à lui la technique du boosting.

Ensuite, le modèle que l'on vient d'apprendre est appliqué pour étiqueter les données d'un autre ensemble pour lequel la liste des étiquettes est déjà connue. Puis, afin de pouvoir réaliser une analyse, nous extrayons des fichiers uniquement les informations qu'il est nécessaire de conserver, c'est à dire les concepts. Nous obtenons ainsi la série des concepts références : la liste des étiquettes, déjà présente dans le corpus de données pour le traitement des langues, que nous venons de ré-étiqueter et la série des concepts hypothèses : la liste des étiquettes que le modèle a attribué à l'ensemble de données analysé.

Enfin, le fichier contenant les deux séquences de concepts est transmis à un analyseur qui repère les alignements des séquences des étiquettes reconnues et des étiquettes références. Cet analyseur compte les suppressions, insertions et substitutions qui servent de mesure d'évaluation. Les oublis ou suppressions ont lieu lorsque le modèle n'a pas placé d'étiquette alors qu'il aurait du en repérer une. Les insertions ou ajouts sont le phénomène contraire des oublis, à savoir qu'une étiquette a été localisée alors qu'il n'y avait rien à détecter. Et finalement les substitutions qui consiste simplement à se tromper d'étiquette.

Par exemple, l'une des substitutions les plus courantes dans les ensembles traités ici est le fait de remplacer une organisation par un lieu, ou vice-versa, comme notamment pour les noms de pays.

En réalité, selon le contexte, il est très difficile pour les algorithmes de différencier deux exemples tels que : « La France a adopté une nouvelle loi » et « La France a accueilli les Jeux Olympiques ». Dans la première phrase, « La France » est une organisation, et dans la seconde, c'est un lieu (Les Jeux Olympiques ont eu lieu à tel endroit). Mais dans les deux phrases, « La France » est sujet du verbe, le terme a donc la même étiquette morpho-syntaxique et la même fonction grammaticale ce qui rend ce genre d'erreurs difficile à retrouver.

2.2 Différentes versions des corpus

Afin de pouvoir identifier des combinaisons d'indices statistiques fiables dans le but de diagnostiquer les annotations erronées, il est nécessaire de disposer de plusieurs moutures des données. Seuls les corpus ATIS et ESTER furent utilisés pour réalisés les tests, le corpus MEDIA quant à lui ne fut pas traité. Nous avons à notre disposition différentes versions de chacun des corpus. Pour le corpus ATIS et le corpus ESTER, nous disposons d'une version brouillon et d'une version corrigée (même si elle n'est pas parfaite).

À l'aide de ces 2 versions des ensembles de données, il est possible de connaître quelles sont les phrases contenant des annotations incorrectes. Grâce à ces 2 variantes pour chaque corpus, nous déterminons ainsi quelles sont les phrases que les critères de sélection doivent repérer et tenter de corriger.

Voici les résultats détaillés obtenus lorsque l'on teste deux systèmes entraînés sur les deux premières versions des ensembles sur leur ensemble de test respectifs :

TAB. 2.1 – Comparaison des performances du système entraîné sur la version originale et sur la version corrigée de l'ensemble de données pour le corpus ATIS

Données		Ensemble original	Ensemble corrigé
Phrases	Taux d'erreur	22,4	13
	Substitutions	18,8	9,2
	Suppressions	1,8	1,8
	Insertions	3,8	3,7
Mots	Taux d'erreur	11,1	6,5
	Substitutions	8,8	4,3
	Suppressions	0,6	0,6
	Insertions	1,7	1,7
Précision		88,9	93,5

TAB. 2.2 – Comparaison des performances du système entraîné sur la version originale et sur la version corrigée de l'ensemble de données pour le corpus ESTER

Données		Ensemble original	Ensemble corrigé
Phrases	Taux d'erreur	31,3	26,8
	Substitutions	9,2	9
	Suppressions	23,5	18,8
	Insertions	2,2	2,1
Mots	Taux d'erreur	28	22,8
	Substitutions	6,9	6,4
	Suppressions	19,6	14,9
	Insertions	1,4	1,5
Précision		72	77,2

Nous constatons ici qu'au niveau du taux d'erreur mots, l'écart de performances du système entraîné avec la version originale et la version corrigée est de 4,6 % pour le corpus ATIS et de 5,2 % pour le corpus ESTER. Pour ce premier tableau de résultats, nous pouvons voir le détail des substitutions, insertions et oublis. Pour ATIS, ceux sont les substitutions qui sont présentes en grand nombre comparés aux suppressions et insertions. Sur l'évaluation avec l'apprentissage à l'aide de l'ensemble original, il y a presque une phrase sur 5 où une étiquette est remplacé par une autre.

Dans le corpus ESTER, ce sont les suppressions qui sont les plus importantes. Presque une étiquette sur 5 est tout simplement oubliée lorsque le test est réalisé à l'aide d'un apprentissage sur l'ensemble original. Mais ce chiffre perd 5 %, il passe à 14,9 % d'oublis d'étiquettes lorsque le test est réalisé avec un apprentissage sur l'ensemble corrigé.

Grâce à ces 2 versions des données, nous avons pu déterminer quelles sont les phrases contenant des annotations incorrectes dans l'ensemble original. En effet, nous nous basons sur l'hypothèse que les phrases ayant des annotations différentes entre les 2 versions du corpus sont les phrases qui étaient fausses dans l'ensemble de départ (l'ensemble original) et qui ont été corrigées. Les 2 tableaux suivants montrent les résultats obtenus à l'aide des 2 ensembles de départ desquels les phrases déclarées fausses ont été retirées.

TAB. 2.3 – Comparaison du modèle appris sur l'ensemble moins les erreurs avec les 2 modèles précédents pour le corpus ATIS

Données	Ensemble original	Ensemble moins erreurs	Ensemble corrigé
Taux d'erreur phrases	22,4	13,1	13
Taux d'erreur mots	11,1	6,6	6,5
Précision	88,9	93,4	93,5

TAB. 2.4 – Comparaison du modèle appris sur l'ensemble moins les erreurs avec les 2 modèles précédents pour le corpus ESTER

Données	Ensemble original	Ensemble moins erreurs	Ensemble corrigé
Taux d'erreur phrases	31,3	27	26,8
Taux d'erreur mots	28	23,3	22,8
Précision	72	76,7	77,2

Pour les ensembles ATIS, l'ensemble de base duquel on a retiré les erreurs fait presque aussi bien que l'ensemble corrigé : suppressions, substitutions et insertions cumulées donnent au total 6,6 % d'erreurs pour les mots pour l'ensemble moins les erreurs et l'ensemble corrigé fait quant à lui 6,6 % d'erreurs.

Par contre, pour les modèles appris sur les ensembles ESTER, il y a environ 1 point d'écart, l'ensemble moins les erreurs n'arrive pas à faire aussi bien que l'ensemble corrigé. En effet, les phrases que l'on a retirées car elles sont considérées comme fausses ne sont peut être pas forcément complètement erronées. Assurément, les phrases des corpus contiennent différents concepts annotés et la plupart du temps, plusieurs concepts sont présents par phrase. Une phrase contenant au moins une différence peut être retirée alors qu'elle contient d'autres informations, qui elles ont été correctement annotées, et sont donc utiles à l'apprentissage. En réalité, les phrases que l'on a retirées contiennent bien évidement du bruit qui trouble l'apprentissage mais aussi de l'information profitable qui est parfois présente uniquement à cet endroit du corpus. Et si cette information est perdue, alors la qualité d'annotation s'en trouve détériorée.

Ce phénomène explique pourquoi il est intéressant de corriger les annotations au lieu de simplement retirer les éléments inexacts.

2.3 Résultats de référence

Grâce aux phrases que l'on a pu déterminer comme fausses, il est possible de définir une *baseline*. En utilisant le seuil de confiance de Wapiti qui effectue ses calculs à l'aide de champs conditionnels aléatoires, une probabilité que la phrase soit correctement annotée peut être obtenue. Cette probabilité est calculée globalement sur l'ensemble des mots de la phrase. Le calcul à l'aide des champs conditionnels aléatoires fournit ainsi une mesure de confiance sur sa propre annotation. De plus, comme déjà cité précédemment, les champs conditionnels aléatoires sont pour l'instant le modèle état de l'art pour ce type de problèmes d'apprentissage car ils utilisent des milliers de caractéristiques afin de prendre une décision sur l'étiquette courante. Enfin, cette probabilité a été utilisé efficacement en apprentissage actif et en détection d'erreurs [10].

Afin d'avoir une première idée de la validité de cette mesure de confiance, un calcul de résultats de référence est réalisé pour chacun des corpus version originale. Ce calcul détermine le nombre de phrases fausses détectées en fonction d'un échantillonnage des probabilités des champs conditionnels aléatoires. Les deux tableaux suivants fournissent

les résultats des calculs de précision, rappel et F-mesure d'abord pour ATIS, puis pour ESTER.

TAB. 2.5 – Évaluation du seuil de confiance pour le corpus ATIS

Seuil	Nombre de phrases sélectionnées	Pourcentage de l'ensemble sélectionné	Précision	Rappel	F-mesure
0,05	7	0,14	57,14	0,89	1,76
0,1	35	0,70	45,71	3,57	6,83
0,15	58	1,17	36,21	4,69	8,30
0,2	105	2,11	37,14	8,71	14,10
0,25	164	3,29	34,76	12,72	18,63
0,3	211	4,24	36,02	16,96	23,07
0,35	267	5,36	32,58	19,42	24,34
0,4	334	6,71	30,24	22,54	25,83
0,45	428	8,60	27,57	26,34	26,94
0,5	532	10,69	25,00	29,69	27,14
0,55	653	13,12	22,66	33,04	26,88
0,6	766	15,39	21,28	36,38	26,85
0,65	916	18,40	20,96	42,86	28,15
0,7	1 093	21,96	20,04	48,88	28,42
0,75	1 340	26,92	18,36	54,91	27,52
0,8	1 564	31,42	17,33	60,49	26,94
0,85	1 857	37,30	15,99	66,29	25,77
0,9	2 214	44,48	14,45	71,43	24,04
0,95	2 758	55,40	13,56	83,48	23,33

TAB. 2.6 – Évaluation du seuil de confiance pour le corpus ESTER

Seuil	Nombre de phrases sélectionnées	Pourcentage de l'ensemble sélectionné	Précision	Rappel	F-mesure
0,05	297	0,58	61,95	1,91	3,71
0,1	652	1,27	57,98	3,93	7,37
0,15	1 068	2,08	56,65	6,29	11,33
0,2	1 626	3,16	55,41	9,37	16,04
0,25	2 246	4,37	53,12	12,41	20,12
0,3	2 921	5,68	51,80	15,74	24,15
0,35	3 649	7,10	50,89	19,32	28,01
0,4	4 527	8,81	49,81	23,46	31,90
0,45	5 485	10,67	48,86	27,88	35,51
0,5	6 608	12,85	48,18	33,13	39,26
0,55	7 926	15,42	47,29	39,00	42,74
0,6	9 287	18,06	46,17	44,62	45,38
0,65	10 768	20,94	44,74	50,13	47,28
0,7	12 461	24,24	43,58	56,50	49,20
0,75	14 544	28,29	42,18	63,83	50,80
0,8	16 716	32,51	40,48	70,41	51,41
0,85	19 119	37,19	38,68	76,95	51,49
0,9	22 288	43,35	36,17	83,87	50,54
0,95	27 181	52,87	32,54	92,02	48,08

Afin de pouvoir déchiffrer ces nombres, il est important de savoir que l'ensemble ATIS est bien plus petit que l'ensemble ESTER. En effet, le corpus ATIS, contient en tout et pour tout 4 978 énoncés. Alors que l'ensemble ESTER comporte quant à lui 51 412 phrases. L'ensemble ESTER est donc 10 fois plus grand. Cependant, le nombre de concepts présents dans ESTER a été réduit à 16. Les concepts présents dans ESTER sont les suivantes : montant, fonction, lieu, organisation, personne, production humaine, temps et inconnu. Pour chacun de ces concepts, il y a deux étiquettes qui lui sont associés : concept-*B* et concept-*I*, cet encodage permet de savoir si le mot analysé se trouve au début ou au milieu du concept (voir schéma 1.3 page 13). Pour l'ensemble ATIS, il y a plus de 80 concepts qui annotés. Ces concepts correspondent aux éléments que l'on a besoin de retrouver dans la base afin de pouvoir formuler une requête tels que la ville et l'heure de départ, l'endroit et le jour de l'arrivée, le nom ou le numéro de la compagnie aérienne, la classe (première ou économique), etc.

Parmi les 4 978 dialogues de l'ensemble ATIS, 448 ont été décelés comme mal annotés, cela représente environ 9 % de la taille de l'ensemble. Dans cet ensemble, la mesure de confiance moyenne est de 0,83, ce qui indique que les champs conditionnels aléatoires sont plutôt confiants dans leur étiquetage. Malgré cela, la F-mesure, qui indique la propension avec laquelle nous sommes capables de retrouver les phrases fausses, n'atteint pas les 30 %.

Effectivement, nous constatons que si l'on se sert de la mesure de confiance des champs conditionnels aléatoires comme seuil de confiance, la précision avec laquelle les phrases fausses sont retrouvées a naturellement tendance à diminuer alors que le seuil de confiance lui augmente. Le rappel des phrases fausses a quant à lui tendance à suivre la même évolution croissante que le pourcentage de phrases sélectionnées. En effet, plus le nombre de dialogues sélectionnés est grand, et plus les possibilités que ces dialogues contiennent des phrases incorrectes progresse.

Alors que la précision ne cesse de s'affaiblir et le rappel de s'accroître, la meilleure F-mesure est atteinte pour un seuil de 0,7, alors qu'environ un cinquième de l'ensemble est sélectionné, elle est de 28,42.

Pour l'ensemble ESTER, nous avons détecté 9 611 énoncés faux sur 51 412 énoncés au total. Cela représente environ 17 % de l'ensemble, le corpus ESTER comporte donc environ deux fois plus de phrases fausses en proportion. Tout comme pour ATIS, la probabilité moyenne des dialogues est bonne, elle est de 0,82. La précision et le rappel suivent les mêmes tendances, mais les chiffres sont tout de même globalement meilleurs. Par exemple, pour le premier échantillon où les phrases sélectionnées sont celles ayant une probabilité globale pour la phrase inférieur à 0.05, la précision dépasse les 60 %, bien évidemment, cela représente très peu de phrases, même pas 1 % de l'ensemble est sélectionné, mais cela montre qu'il est possible avec des critères assez sélectifs de retrouver les erreurs.

De plus, il est normal que la précision ait tendance à diminuer puisque plus le seuil de confiance est bon, plus les champs conditionnels sont sûrs de leur annotation. Donc plus les phrases sélectionnées ont une mesure de confiance élevée et plus elles ont de chance d'être correctes.

Comme les chiffres sont globalement meilleurs pour ESTER, la F-mesure la plus éle-

vée est bien supérieure, elle dépasse les 50 %. De fait, elle est atteinte pour un seuil plus élevé que dans le corpus ATIS, puisqu'il fait atteindre un seuil de confiance de plus 0,8 pour voir la F-mesure parvenir à un peu plus de 51 %.

Maintenant que l'on possède une évaluation de référence pour le calcul pour les deux corpus, et que l'on a un seuil de confiance initial qui permet d'avoir une première idée de ce que donne une mesure de confiance basique, le but va être de découvrir des indices statistiques afin d'améliorer la détection des phrases erronées.

Chapitre 3

Travail réalisé

Afin d'améliorer la détection des erreurs, nous allons utiliser différents types d'indices statistiques connus dans la littérature, soumettre l'utilisation d'autres critères et surtout utiliser ces indices de manière conjugués.

Tout d'abord, commençons par rappeler les indices généralement utilisés. En premier lieu, nous pouvons citer les machines à vecteurs de support [2] et [12]. En effet, les SVM sont une méthode historique, elles sont parmi les premières à avoir donné des résultats en traitement automatique du langage naturel et en particulier en décodage conceptuel. Même si elles ne prennent pas en compte l'ordre des mots, elles sont tout de même adaptées à ce type de tâche car elles permettent de gérer de grandes quantités de données.

Ensuite, quelques dizaines d'années plus tard, les modèles de Markov cachés adaptés pour le traitement des graphes de mots, nommés transducteurs à états fini sont apparus [3] et [2]. Ces transducteurs se basent sur le calcul de la probabilité *a posteriori* de la séquence d'étiquettes en fonction de la séquence de mots analysés. L'avantage de cette manière de faire est que l'algorithme est moins sensible aux erreurs puisq'elles sont gommées par le calcul des probabilités. Une méthode à base de transducteurs ne réannotera donc pas forcément à l'identique une étiquette erronée.

Puis, intéressons nous maintenant au modèle état de l'art pour le moment pour le traitement des langues, à savoir les champs conditionnels aléatoires [2] et [4]. Les champs conditionnels aléatoires ont déjà été utilisés pour la détection et la correction d'erreur en traitement automatique des langues [10]. Les champs conditionnels ont généralement l'avantage contrairement aux méthodes qui utilisent la technique du sac de mots de prendre en compte l'ordre des vocables. Pour prendre une décision, ils se basent sur une fenêtre plus ou moins grande définie autour du mot courant (bigrammes, trigrammes).
Malheureusement, les champs conditionnels aléatoires apprennent tellement bien qu'ils apprennent même les erreurs d'annotation. En effet, lorsque l'on fait apprendre les champs conditionnels aléatoires sur un ensemble corrigé et que l'on réannote cet ensemble avec le modèle tout juste appris, ce modèle réannote à l'identique plus de 99 % de l'ensemble.

Attardons nous à présent sur un processus généralement moins communément utilisé mais tout de même parfois appliquée en traitement des langues, le boosting. Le principe

du boosting est le suivant, il consiste à se concentrer sur les exemples de dialogues les plus compliqués afin d'extraire des informations les meilleures possibles [12]. Cette technique utilise des règles de classification qui vote chacune à leur tour, et la classe ayant reçu le plus de voix l'emporte.

L'un des principaux intérêts du boosting est que ce type d'algorithmes permet d'associer plusieurs concepts à un même mot. Seulement c'est une pratique qui prend du temps car elle nécessite un grand nombre de lectures des données et il est difficile de déterminer combien de passages sur les données vont être nécessaires.

Enfin, en dernier lieu, considérons les arbres de décisions sémantiques [12]. Parmi les 5 modèles d'apprentissage exposés ici, c'est la seule méthode qui n'avait pas été introduite dans l'état de l'art car peu de documents en parlent et elle n'est que rarement utilisée pour ce genre de tâches d'étiquetage. Ce procédé a, de la même manière que le boosting, recours à des règles de classification statistiques. En fonction des données disponibles, différentes questions sont proposées à chaque nœud de l'arbre, et la question qui a le meilleur score l'emporte, c'est elle qui est choisie. Les principaux bénéfices des arbres de décision sémantique sont qu'ils se basent sur un vocabulaire restreint, et qu'ils prennent en compte l'ordre des termes.

3.1 Descripteurs utilisés

Voyons maintenant la liste des descipteurs utilisés pour déterminer des critères efficaces.

1) longueur : permet de connaître le nombre de mots de la phrase.

2) probabilité forte : probabilité marginale la plus forte pour la phrase.

3) probabilité faible : probabilité marginale la plus faible pour la phrase.

4) probabilité forte concepts : probabilité marginale la plus forte pour les concepts reconnus (concepts autres que null).

5) probabilité faible concepts : probabilité marginale la plus faible pour les concepts reconnus.

6) probabilité globale : probabilité globale de la phrase.

7) probabilité moyenne : probabilité moyenne pour tous les concepts de la phrase.

8) moyenne concepts : probabilité moyenne pour les concepts reconnus.

9) nombre de concepts : nombre de concepts reconnus dans la phrase.

10) globale/longueur : rapport entre la probabilité globale de la phrase et le nombre de mots de cette phrase. En effet, plus la phrase est longue, et plus la probabilité globale risque d'être faible. Le rapport globale/longueur a donc plus de chances d'être indicatif que la probabilité globale et la longueur seules.

11) poids boosting : booléen qui indique si la phrase courante a un poids important avec le boosting ou pas. Pour cet indicateur, nous gardons uniquement les 10 % de poids les plus importants.

12) réannotation CRF : booléen qui indique si la phrase courante est réannotée à l'identique par les CRF ou pas.

13) concaténation boosting-CRF : poids boosting - réannotation CRF (4 possibilités : vrai-vrai, vrai-faux, faux-vrai, faux-faux).

14) OU exclusif : OU exclusif entre le boosting et les CRF.

15) OU (logique) : OU logique entre le boosting et les CRF.

16) ET (logique) : ET logique entre le boosting et les CRF.

17) réannotation arbre : booléen qui indique si la phrase courante est réannotée à l'identique par les arbres ou pas.

18) réannotation fsm : booléen qui indique si la phrase courante est réannotée à l'identique par les modèles de Markov ou pas.

19) réannotation svm : booléen qui indique si la phrase courante est réannotée à l'identique par les machines à vecteurs de support ou pas.

20) nb modèles : nombre de modèles capables de réannoter à l'identique la phrase courante.

Afin d'effectuer un tri parmi ces différents descripteurs, un arbre de décision est utilisé. Le critère avec lequel l'arbre est calculé est la minimisation de l'erreur. En effet, ce que nous recherchons ici est non pas une notion d'ordre mais le moins d'erreurs de classification possible. Malheureusement, les différents tests réalisés à l'aide de ces descripteurs n'ont rien donné de concluant. L'arbre n'a pas réussi à trouver des critères suffisamment robustes qui permettrait de détecter les phrases fausses efficacement.

La plupart des ces descripteurs sont basés sur les données fournies par les CRF, mais nous avons à notre disposition 5 modèles d'apprentissage, aussi regardons plus en détails ce que donne ces modèles.

3.2 Expériences avec les modèles individuels

Dans le but d'identifier des critères pertinents, nous commençons d'abord par regarder ce que valent les 5 modèles individuellement. Pour chacun des modèles, l'apprentissage est réalisé à l'aide de l'ensemble original, et le modèle tout juste appris est testé sur l'ensemble sur lequel il a été entraîné. Le modèle est évalué sur le même ensemble que l'ensemble d'apprentissage car cela permet de voir quelles phrases sont réannotées à l'identique. Ensuite, les phrases qui justement ne sont pas réannotées à l'identique sont extraites. Si le modèle ne les réannote pas comme l'annotation de référence déjà présente dans le corpus, alors il y a des risques que cette annotation de référence soit incorrecte, étant donné que le modèle ne l'a pas reproduite.

Pour les modèles appris à base d'arbres, de transducteurs, de machines à vecteurs de support ou à l'aide de Wapiti, les expériences ont été réalisées avec les phrases que chacun de ces modèles ne réannotent pas de la même manière. Par contre, pour BonzaiBoost, étant donné que c'est une technique à base de boosting qui est utilisée, ce sont les exemples ayant les poids les plus importants qui sont extraits du corpus et utilisés pour réaliser les tests.

Les résultats suivants montrent ce que donnent les 5 modèles d'apprentissage sur le corpus ATIS, l'ensemble ESTER ainsi que sur le regroupement des deux ensembles de données.

TAB. 3.1 – Résultats des 5 modèles d'apprentissage entraînés sur l'ensemble original pour ATIS

Modèle d'apprentissage	Nombre de phrases sélectionnées	Vrais positifs	Précision	Rappel	F-mesure
Wapiti	340	137	40,29	30,58	34,77
Boosting	99	52	52,53	11,61	19,01
Arbre	717	265	36,96	59,15	45,49
FSM	402	182	45,27	40,63	42,82
SVM	189	73	38,62	16,29	22,92

TAB. 3.2 – Résultats des 5 modèles d'apprentissage entraînés sur l'ensemble original pour ESTER

Modèle d'apprentissage	Nombre de phrases sélectionnées	Vrais positifs	Précision	Rappel	F-mesure
CRF	1 121	783	69,85	8,15	14,59
Boosting	15 743	5 839	37,09	60,75	46,06
Arbre	14 875	5 735	38,55	59,67	46,84
FSM	5 508	2 802	50,87	29,15	37,07
SVM	3 131	1 501	47,94	15,62	23,56

TAB. 3.3 – Résultats des 5 modèles d'apprentissage entraînés sur le regroupement des ensembles d'origine pour ATIS et ESTER

Modèle d'apprentissage	Nombre de phrases sélectionnées	Vrais positifs	Précision	Rappel	F-mesure
CRF	1 461	920	62,97	9,15	15,97
Boosting	15 842	5 891	37,19	58,56	45,49
Arbre	15 592	6 000	38,48	59,65	46,78
FSM	5 910	2 984	50,49	29,66	37,37
SVM	3 320	1 574	47,41	15,65	23,53

Comme nous pouvons le constater, les résultats correspondent à ce qu'il est possible de trouver dans la littérature.

Pour l'ensemble ATIS, la précision maximale obtenue est de 52 % environ grâce au boosting, ce qui signifie que parmi les phrases ayant un poids de boosting important, un

peu plus d'une sur deux est fausse.

Si l'on prend en compte à la fois la précision de repérage des phrases fausses parmi les phrases sélectionnées comme potentiellement fausses, et le rappel qui indique le nombre de phrases effectivement fausses trouvées parmi l'ensemble des phrases du corpus, alors le meilleur modèle quelque soit le jeu de données étudié, ce sont les arbres de décisions sémantiques. En effet, dans les 3 tableaux, la meilleure F-mesure est obtenue à chaque fois lorsque les tests sont réalisés avec des arbres.

Même si pour l'ensemble ESTER, le rappel est légèrement plus élevé avec le boosting, la précision diminue un peu, ce qui donne des F-mesures équivalentes.

À noter que pour l'ensemble ESTER qui est bien plus grand que l'ensemble ATIS, ce sont les CRF qui obtiennent la meilleure précision, quasiment 70 %. Mais forcément, en contrepartie, cela détériore le rappel qui n'atteint même pas les 10 %.

Dans les résultats concernant le regroupement des deux ensembles, ESTER étant dix fois plus grand qu'ATIS, son poids est beaucoup plus important, les chiffres suivent donc la même tendance.

3.3 Expériences avec différentes combinaisons de modèles

Maintenant que les résultats de chacun des modèles testé individuellement sont connus, le but de cette partie va être de montrer comment nous pouvons les combiner entre eux afin d'obtenir de meilleurs scores.

Afin d'éviter de tester toutes les combinaisons imaginables qu'il est possible de réaliser avec les 5 modèles, un arbre de décision est utilisé afin de réaliser un premier tri. À l'aide de cet arbre, nous cherchons à identifier les combinaisons d'indices les plus pertinentes qui fonctionnent sur les deux corpus. Les tableaux suivants montrent une partie des tests réalisés à l'aide de différentes combinaisons de modèles pour lesquelles des résultats, généralement supérieurs à ceux obtenus avec les modèles individuels, ont été produits.

Les tableaux 3.4 et 3.5 présentent les résultats par ordre croissant de modèles exploités.

Pour la première ligne du tableau 3.4 par exemple, il n'y a que 78 de phrases de sélectionnées parmi les 4 978 phrases de l'ensemble ATIS car seules les phrases à la fois mal réannotées par les CRF et ayant un important poids de boosting sont sélectionnées. Pour les deux ensembles, la combinaison Boosting - CRF donne une précision élevée, ces 2 modèles combinés sont donc efficaces lorsque le but est d'éviter qu'il y ait trop de bruit. Par contre, la précision étant bonne, le rappel est bien évidemment assez faible, un peu plus de 10 % pour ATIS et environ 7 % pour ESTER.

Pour l'ensemble ATIS, c'est la combinaison Boosting - FSM qui obtient la meilleure précision, cela peut s'expliquer par le fait que cet ensemble est beaucoup plus simple que le corpus ESTER, le bosting combiné à des modèles de Markov est donc très efficace pour

Tab. 3.4 – Résultats de différentes combinaisons de modèles entraînés sur l'ensemble original pour ATIS

Combinaison de modèles	Nombre de phrases sélectionnées	Vrais positifs	Précision	Rappel	F-mesure
Boosting CRF	78	49	62,82	10,94	18,63
CRF Arbre	296	135	45,61	30,13	36,29
CRF FSM	215	109	50,70	24,33	32,88
Arbre FSM	248	129	52,02	28,79	37,07
Boosting Arbre	82	50	60,98	11,16	18,87
Boosting FSM	71	46	64,79	10,27	17,73
Boosting SVM	64	41	64,06	9,15	16,02
Arbre SVM	155	73	47,10	16,29	24,21
FSM SVM	119	59	49,58	13,17	20,81
CRF SVM	140	69	49,29	15,40	23,47
CRF FSM Arbre	196	107	54,59	23,88	33,23
CRF FSM Arbre SVM	111	59	53,15	13,17	21,11

repérer les phrases contenant au moins une annotation incorrecte. Le boosting combiné aux machines de vecteurs de support obtient à peu près les mêmes résultats, ce qui montre qu'en combinant seulement 2 modèles, plus de 10 points de précision ont pu être gagnés pour un rappel à peu près équivalent.

Par contre, dès que l'on commence à combiner les éléments, il y a moins de phrases qui sont sélectionnées donc globalement le rappel diminue. L'association de modèles ayant le rappel le plus important est CRF - Arbre dans le cas présent. Ce rappel vaut 30 %, mais malheureusement la précision pour cette association n'est que de 45 %.

En ce qui concerne la F-mesure, la meilleure que l'on obtient est de 37 % pour le couplage Arbre - FSM. C'est pour ces modèles que l'équilibre précision-rappel est le plus stable. En effet, le rappel se rapproche de 29 % et la précision est de plus de 50 %.

Les résultats de l'ensemble ESTER sont quant à eux légèrement différents. Effectivement, les meilleures précisions sont obtenues sont lorsqu'un nombre plus important de modèles est groupé. Les associations CRF - FSM, CRF - SVM et CRF - FSM - Arbre obtiennent des résultats préférables au niveau de la précision, mais en contre partie, leur rappels est toujours à environ 6 %. Pour cet ensemble, le meilleur rappel obtenu est de quasiment 50 % pour le groupement Bonzai - Arbre, ce qui peut s'expliquer par le fait ceux sont les 2 modèles qui individuellement sélectionne le plus de phrases. Quant à la précision, les 75 % sont presque atteints lorsque 4 modèles sont utilisés conjointement, les CRF, les FSM, les arbres et les SVM. Le gain est donc de 5 points par rapport au modèle individuel des CRF, mais cela provoque une perte de 3 % de rappel et de 5 % de F-mesure.

Tab. 3.5 – Résultats de différentes combinaisons de modèles entraînés sur l'ensemble original pour ESTER

Combinaison de modèles	Nombre de phrases sélectionnées	Vrais positifs	Précision	Rappel	F-mesure
Boosting CRF	932	661	70,92	6,88	12,54
CRF Arbre	988	706	71,46	7,35	13,32
CRF FSM	873	635	72,74	6,61	12,11
Arbre FSM	4 164	2 345	56,32	24,40	34,05
Boosting Arbre	11 641	4 778	41,04	49,71	44,97
Boosting FSM	3 977	2 194	55,17	22,83	32,29
Boosting SVM	2 539	1 264	49,78	13,15	20,81
Arbre SVM	2 428	1 260	51,89	13,11	20,93
FSM SVM	1 484	929	62,60	9,67	16,75
CRF SVM	888	640	72,07	6,66	12,19
CRF FSM Arbre	817	593	72,58	6,17	11,37
CRF FSM Arbre SVM	659	493	74,81	5,13	9,60

3.4 Expériences en fonction du nombre de modèles qui réannote correctement

Après avoir expérimenté différents regroupements de modèles, nous avons décidé de tester une méthode différente où les modèles ne sont pas reconnus individuellement. En effet, avec cette manière de faire, au lieu d'identifier quel modèle réannote correctement ou pas, nous comptons simplement le nombre de modèles qui ne réannote pas la phrase courante à l'identique.

Étant donné que nous avons à notre disposition 5 modèles d'apprentissage, les calculs sont réalisés en partant de 5 modèles qui réannote à l'identique la phrase, puis 4, puis 3...jusqu'à ce qu'aucun des modèles ne sache plus réannoter. Les résultats de ces expériences sont présentés dans les tableaux 3.6 à 3.8.

Bien évidemment, plus le nombre de modèles qui réannote correctement est faible, et plus les chances de trouver des phrases mal annotées sont importantes. D'ailleurs, nous pouvons constater que plus le nombre de modèles qui réannote à l'identique diminue, et plus la précision augmente. En effet, pour ATIS, nous réussissons à obtenir 67,31 % de précision, soit 3 point de plus que la meilleure précision que nous possédions jusqu'à présent. En gagnant en précision, le rappel diminue, il est en dessous de 10 % et la F-mesure passe à 14 %.

Lorsqu'aucun des modèles n'est capable de réannoter la phrase à l'identique, cela représente 52 phrases sélectionnées pour ATIS, donc à peu près 1 % de la taille totale

TAB. 3.6 – Résultats en fonction du nombre de modèles qui réannote correctement sur l'ensemble original pour ATIS

Nombre de modèles qui réannotent correctement	Nombre de phrases sélectionnées	Vrais positifs	Précision	Rappel	F-mesure
5 modèles	4 049	130	3,21	29,02	5,78
4 modèles	526	153	29,09	34,15	31,42
3 modèles	171	47	27,49	10,49	15,19
2 modèles	101	45	44,55	10,04	16,39
1 modèle	79	38	48,10	8,48	14,42
0 modèle	52	35	67,31	7,81	14,00

TAB. 3.7 – Résultats en fonction du nombre de modèles qui réannote correctement sur l'ensemble original pour ESTER

Nombre de modèles qui réannotent correctement	Nombre de phrases sélectionnées	Vrais positifs	Précision	Rappel	F-mesure
5 modèles	31 250	2 449	7,84	25,48	11,99
4 modèles	6 954	1 704	24,50	17,73	20,57
3 modèles	8 293	2 846	34,32	29,61	31,79
2 modèles	3 419	1 630	47,67	16,96	25,02
1 modèle	899	536	59,62	5,58	10,20
0 modèle	597	446	74,71	4,64	8,74

de l'ensemble. Sur ces 52 phrases sélectionnées, 35 sont des phrases qui ont été déclarées fausses, ce qui équivaut à 7,8 % des phrases fausses (35 sur 448). En proportion, il est vrai que le nombre de phrases erronées repérées est faible mais nous savons qu'elles ont environ 7 chances sur 10 d'être effectivement incorrectes. En sachant cela, il est plus simple soit de les retirer, soit de tenter de les corriger. Et puis, le fait d'avoir une pré-sélection permet d'économiser du temps et de l'énergie qui auraient été gaspillés en calculs inutiles. D'autre part, il est beaucoup plus simple à la fois pour un humain ou pour un programme d'avoir 50 phrases au lieu de 5 000.

Pour l'ensemble ESTER, la précision la meilleure obtenue est à peu près identique à celle obtenue avec la meilleure combinaison de modèles. Par contre, le rappel est un peu plus faible. Mais ce qui important ici, c'est plutôt le fait que même lorsque les 5 modèles réannotent correctement, il y a toujours quasiment 8 % de précision. Alors que les modèles réannotent à l'identique, ce qui en théorie est signe d'annotation cohérente, il reste 2 449 vrais positifs, donc 2 449 phrases mal annotées sur 31 250 phrases sélectionnées. Étant donné que cela semble relativement étrange, d'autres tests ont été effectués.

Au vu du nombre de vrais positifs restants alors que les arbre, les CRF, les FSM et

TAB. 3.8 – Résultats en fonction du nombre de modèles qui réannote correctement sur le regroupement des ensembles d'origine pour ATIS et ESTER

Nombre de modèles qui réannotent correctement	Nombre de phrases sélectionnées	Vrais positifs	Précision	Rappel	F-mesure
5 modèles	35 299	2 579	7,31	25,64	11,37
4 modèles	7 480	1 857	24,83	18,46	21,18
3 modèles	8 464	2 893	34,18	28,76	31,24
2 modèles	3 520	1 675	47,59	16,65	24,67
1 modèle	978	574	58,69	5,71	10,40
0 modèle	649	481	74,11	4,78	8,98

les SVM réannotent avec la même annotation et que le boosting n'indique pas de poids particulier, la probabilité des CRF qui sert de mesure de référence a du coup elle aussi été vérifiée. Le résultat est relativement surprenant puisque parmi les 448 phrases erronées contenues dans ATIS, 91 de ces phrases, soit environ 20 % des phrases erronées ont une probabilité calculée par les CRF de plus 0,9. En théorie, ces phrases sont donc loin d'être fausses puisque tous les indicateurs les placent plutôt dans des catégories tels que très bonne voire excellente.

Ce test a aussi été réalisé sur le corpus ESTER. Cette fois-ci, les résultats sont légèrement meilleurs. En effet, sur les 9 611 phrases erronées que comporte l'ensemble ESTER, seules environ 12 % des phrases incorrectes (soit 1 199 phrases) ont une mesure de confiance de plus de 0,9. Malgré tout, ces phrases mal annotées ne sont absolument pas détectables pour le moment. Dans l'état actuel des choses, elles resteront indiscernables des phrases bien annotées tant que les indicateurs les classeront comme très bonnes.

Conclusion

Le but de notre travail était d'améliorer les techniques de détection automatique d'erreurs d'annotation existantes, afin d'augmenter les performances des algorithmes d'apprentissage automatique.

Dans un premier temps, nous avons testé différents descripteurs pour la plupart issus des données disponibles grâce aux champs conditionnels aléatoires. Ces descripteurs ne furent malencontreusement pas assez robustes pour fournir des critères de détection satisfaisants.
Dans un second temps, nous avons tout d'abord évalués individuellement les différents modèles d'apprentissage mis à notre disposition. Certains de ces modèles étaient déjà connus dans la littérature et nous en avons ajoutés d'autres non traditionnellement utilisés pour ce type de tâche.
Ensuite, nous avons testé différentes combinaisons de ces modèles. Et enfin, nous nous sommes basés sur le nombre de modèles réannotant à l'identique la phrase courante pour réaliser nos calculs.

En conclusion, nous avons finalement obtenu un point de fonctionnement avec 0 modèles qui réannotent correctement la phrase courante, meilleur que les résultats de référence, et cela pour les 2 corpus.

Grâce à ces résultats, plusieurs perspectives sont envisageables. En premier lieu, nous pouvons réfléchir à l'éventualité d'éliminer les phrases déclarées fausses et de réapprendre tous les modèles. Ensuite, avec les mêmes critères de détection, nous pouvons tester si il est à nouveau possible de détecter des phrases mal annotées.

L'autre perspective envisageable est de détecter peu de phrases fausses mais de manière très précise. Il sera ensuite possible de corriger ces phrases à la main ou à l'aide d'un programme et de réapprendre les modèles. Puis, parmi les nouvelles phrases non réannotés à l'identique par les modèles réappris, il sera peut être possible de détecter d'autres phrases fausses.

De plus, il est aussi possible d'utiliser les descripteurs d'une autre manière ou bien d'autres types de descripteurs tels que des n-grammes par exemple.

Enfin, pour terminer, nous pouvons très bien imaginer de prendre le problème à l'envers. Même si comme nous l'avons vu, cette manière de faire semble plus ardue, il est concevable de sélectionner les phrases dont nous sommes sûrs de l'exactitude, et de tenter de corriger le reste.

Bibliographie

[1] Raymond C. & Fayolle J., 2010. Reconnaissance robuste d'entités nommées sur de la parole transcrite automatiquement. *TALN 2010*, Montréal, Canada.

[2] Raymond C. & Riccardi G., 2007.Generative and discriminative algorithms for spoken language understanding. *International Conference on Speech Communication and Technologies*, Anvers, Belgium.

[3] Raymond C., Béchet F., De Mori R. & Damnati G., 2006.On the use of finite state transducers for semantic interpretation.*Speech Communication*, vol. 48 (3-4), pages 288-304.

[4] Lafferty J., McCallum A. & Pereira F., 2001. Conditional random fields : probabilistic models for segmenting and labeling sequence data. *ICML*, pages 282–289.

[5] Dahl D., Bates M., Brown M., Fisher W., Hunicke-Smith K., Pallett D., Pao C., Rudnicky A. & Shriberg E., 1994. Expanding the scope of the ATIS task : the ATIS-3 corpus. *HLT*, pages 43-48.

[6] Bonneau-Maynard H., Rosset S., Ayache C., Kuhn A., Mostefa D. & MEDIA consortium, 2005. Semantic annotation of the french MEDIA dialog corpus. *InterSpeech*, Lisbonne, Portugal.

[7] Servan C. & Béchet F., 2006. Décodage conceptuel et apprentissage automatique : application au corpus de dialogue Homme-Machine MEDIA. *TALN 2006*, Louvain, Belgique.

[8] Gravier G., Bonastre J-F., Geoffrois E., Galliano S., Mc Tait K. & Choukri K., 2004. ESTER, une campagne d'évaluation des systèmes d'indexation automatique d'émissions radiophoniques en français. *Journées d'Etude sur la Parole*.

[9] Conventions et plans d'évaluation des campagnes ESTER : `http://www.afcp-parole.org/ester/docs.html`

[10] Raymond C. & Riccardi G., 2008. Learning with noisy supervision for spoken language understanding. *Proceedings of the International Conference on Acoustic Speech and Signal Processing*, pages 4989-4992, Las Vegas, États-Unis.

[11] Raymond C., Béchet F., De Mori R. & Damnati G., 2004. Stratégie de décodage conceptuel pour les applications de dialogue oral. *XXVième Journées d'Études sur la parole (JEP)*, Fès, Maroc.

[12] Camelin N., 2004. Décodage conceptuel et classification automatique dans les applications de dialogue oral téléphoniques.

[13] Vlachos A., 2006. Active Annotation. *Proceedings of the workshop on Adaptive Text Extraction and Mining at EACL*, Trente, Italie.

[14] Tur G., Rahim M. & Hakkani-Tür D., 2003. Active labeling for spoken language understanding. *Eurospeech*, Genève, Suisse.

[15] Eskin E., 2000. Detecting errors within a corpus using anomaly detection.