Evaluation systématique d'une méthode commune de génération

Hugo Boulanger

Actes de la 6e conférence conjointe Journées d'Études sur la Parole (JEP, 31e édition),
Traitement Automatique des Langues Naturelles (TALN, 27e édition),
Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL, 22e édition)
Nancy, France, 08-19 juin 2020

Évaluation systématique d'une méthode commune de génération

Hugo Boulanger[1]
(1) Université Paris-Saclay, LIMSI, Campus Universitaire bâtiment 507, Rue du Belvédère, 91400 Orsay, France
prénom.nom@limsi.fr

RÉSUMÉ

Avec l'augmentation de l'utilisation du traitement automatique des langues arrivent plusieurs problèmes dont l'absence de données dans les nouveaux domaines. Certaines approches d'apprentissage tel que l'apprentissage zero-shot ou par transfert tentent de résoudre ces problèmes. Une solution idéale serait de générer des données annotées à partir de bases de connaissances des domaines d'intérêt. Le but de notre travail est d'évaluer une méthode de génération simple et de trouver les critères permettant de la mettre en oeuvre correctement. Pour cela, nous comparons les performances d'un modèle obtenu sur des tâches d'annotation quand il est entraîné sur des données réelles ou sur des données générées. Grâce aux résultats obtenus et à des analyses effectuées sur les données, nous avons pu déterminer des bonnes pratiques d'utilisation de cette méthode de génération sur la tâche d'annotation.

ABSTRACT

Systematic evaluation of a common generation method.

As natural language understanding expands, new domains aim to use this tool. This expansion causes many problems, such as the scarcity of data for these domains. Machine learning approaches such as transfer or zero-shot learning deal with the lack of data, but such methods require labeled data. An ideal solution to this problem would be to generate labeled data from in-domain, highly available data such as ontologies. The goal of our work is to evaluate a simple generation method and to find out criterion to make the method best applicable. Systematic evaluation of the generation method is done by generating and augmenting datasets for sequence labelling benchmarks and comparing the performances obtained from those datasets to real datasets. This paper presents the results of such method and analysis of the original datasets to flesh out guidelines on how to properly use generation to obtain good performances on the associated task.

MOTS-CLÉS : TAL, augmentation de données, génération de données synthétiques.

KEYWORDS: NLP, data augmentation, data generation.

1 Introduction

Le traitement automatique de la langue (TAL) est utilisé dans de plus en plus de domaines et se sert de systèmes d'apprentissage pour résoudre les diverses tâches le composant. Cette recrudescence des systèmes d'apprentissage est accompagnée d'une hausse de la demande de données annotées pour différentes tâches et différents domaines. En revanche, la prise de conscience récente sur l'utilisation des données rend le procédé de récolte de données plus difficile. Dans ce contexte, il devient nécessaire de travailler sur des méthodes permettant d'utiliser des données existantes pour résoudre de nouvelles

Volume 3 : Rencontre des Étudiants Chercheurs en Informatique pour le TAL, pages 43–56.
HAL : hal-02786185.

tâches ou travailler sur de nouveaux domaines. Les exemples les plus communs sont l'utilisation d'apprentissage par transfert (Pan & Yang, 2010; Ruder, 2019) ou d'augmentation (Kafle *et al.*, 2017). Ces techniques ont cependant besoin de données annotées du domaine. Il serait intéressant de pouvoir s'affranchir d'une telle contrainte en générant des données à partir de données plus facilement accessible tel que des bases de connaissances ou des ontologies.

Dans cet article, nous faisons une analyse systématique d'une méthode commune de génération de données annotées. Cette méthode est la méthode de remplissage de patrons. Le principe de cette méthode est d'utiliser des énoncés à trou, les patrons, pour créer de nouveaux exemples en remplissant les trous par des mentions ayant les mêmes concepts que les trous. Cette méthode est illustrée Figure 1. Nous évaluons la performance de la méthode de génération à travers la performance d'un modèle entraîné sur les données générées. C'est ce qui sera appelé performance, sauf indication contraire. Nous comparons les performances entre les modèles entraînés sur les données de référence et les modèles entraînés sur les données générées à partir des mêmes données de références. A notre connaissance, aucune autre étude n'a effectué d'examen systématique de ce type de méthode de génération pour une tâche d'étiquetage. Notre contribution n'est pas la méthode de génération en elle-même, car elle est déjà largement utilisée, mais c'est la connaissance de l'impact des données générées par ces méthodes sur les performances des modèles d'apprentissage, et c'est aussi la connaissance de la meilleure façon d'utiliser cette méthode de génération.

2 État de l'art

Au sein du TAL, des méthodes de génération ont déjà été utilisées pour créer de nouvelles tâches, comme les tâches bAbI (Weston *et al.*, 2016) dans le contexte des systèmes de dialogue qui visaient à fournir des données pour former des modèles de dialogue de bout en bout. Un examen de la qualité de certaines des tâches bAbI a été effectué dans l'article présentant les hybrid code network (Williams *et al.*, 2017). Il s'avère que la méthode de génération utilisée était trop systématique et que de simples systèmes tels que des systèmes à base de règles ont facilement atteint une performance maximale.

Une approche similaire a été étudiée afin de construire un ensemble de données pour entraîner un modèle de compréhension dans le contexte d'un assistant médical (Neuraz *et al.*, 2018). Dans ce travail, la méthode de génération basée sur le remplissage de patrons et une méthode basée sur la paraphrase ont été comparées. Cette étude montre qu'il est possible d'utiliser des méthodes de génération naïves dans le contexte de l'absence de données. Elle souligne également que l'utilisation de la méthode de paraphrase afin d'augmenter les données n'améliore pas de manière significative les performances dans cette tâche.

Des méthodes plus complexes de génération et d'augmentation existent, comme l'utilisation d'auto-encodeurs variationnels (Kingma & Welling, 2014; Yoo *et al.*, 2019) mais elles nécessitent des données étiquetées, ce qui n'est pas conforme à ce que nous visons. Un autre aspect de notre recherche vise à expliquer les résultats que nous obtenons. À cette fin, les jeux de données de références sont analysées dans le but de trouver des indicateurs qui pourraient être liés aux performances des modèles. Liés à ces analyses sont les analyses effectuées dans (Béchet & Raymond, 2019) décrivant la façon dont les modèles réagissent à chacun des benchmarks et établissant un classement de difficulté des dits benchmarks.

3 Tâche de Compréhension

Enoncé	Is	there	a	flight	to	Atlanta
Etiquettes	O	O	O	O	O	B-city

TABLE 1 – Exemple d'un énoncé et de ses étiquettes. Le but de l'étiquetage est de classer chaque token. Les étiquettes ont deux variantes basées sur leur position suivant le format BIO : B pour le début de la mention, I pour l'intérieur de la mention et O pour extérieur à la mention.

Avant d'expliquer la méthode de génération, il est important de comprendre la tâche que nous voulons résoudre. La tâche abordée dans cet article est la tâche d'étiquetage pour la compréhension du langage naturel. Cette tâche vise à classer chaque token d'un énoncé comme expliqué dans le tableau 1. La classification des énoncés ne fera pas partie de notre analyse.

3.1 Etiquetage

L'étiquetage est une tâche de segmentation et de classification. C'est une tâche pour laquelle des méthodes de génération naïves, telles que le remplissage de patrons, sont déjà largement utilisées. Nous avons utilisé cinq ensembles de données connues pour évaluer le potentiel de cette méthode de génération naïve.

3.2 Benchmarks

Les tâches sur lesquelles la génération est évaluée sont les mêmes que celles utilisés par Béchet et Raymond dans leur article Béchet & Raymond (2019) :

a ATIS (Dahl *et al.*, 1994) ou Air Travel Information System est une tâche de réservation de trajet en avion en Anglais. Exemple : "what airlines go to pittsburgh"

b M2M (Shah *et al.*, 2018) ou Machine Talking To Machines est une tâche de réservation de restaurant et de place de cinéma en Anglais. Exemple : "I would like movie tickets to watch avatar ."

c SNIPS (Coucke *et al.*, 2018) est une tâche d'assistant vocal en Anglais. Example : "Look up Applied Linguistics"

d SNIPS70 est l'ensemble des énoncés utilisés pour l'expérience contenant 70 énoncés par classe d'énoncé dans (Coucke *et al.*, 2018). [1]

e MEDIA (Bonneau-Maynard *et al.*, 2005) est une tâche de réservation de chambre d'hôtel en Français. Ce corpus fût construit à partir de transcriptions orales. Exemple : "euh réserver dans cet hôtel". Dans cet article nous utilisons une version légèrement différente de celle utilisée dans (Béchet & Raymond, 2019) car nous utilisons les tours de parole mixés (plusieurs locuteur parlent dans le même énoncé) et les tours de parole vide (aucun élément n'est étiqueté). [2]

1. SNIPS est composé des énoncés des fichiers "train_class_full.json" et SNIPS70 est composé des énoncés des fichier "train_class.json" de https://github.com/snipsco/nlu-benchmark .
2. Nous nous sommes rendu compte de la différence après les expériences.

Corpus	ATIS	M2M	SNIPS	SNIPS70	MEDIA
# tokens du jeu d'entraînement	50497	45965	121547	15172	94708
Vocabulaire du jeu d'entraînement	867	688	13582	3222	2100
# énoncés du jeu d'entraînement	4478	8148	13284	1600	12916
# énoncés du jeu de développement	500	2116	500	500	1259
# énoncés du jeu de test	893	4800	700	700	3518
# concepts	79	12	39	39	73

TABLE 2 – Description des corpus.

Tous les jeux de données n'ont pas été construits de la même manière, et certains n'ont pas inclus de jeu de développement. Nous avons construit des jeux de développement pour les jeux de données qui n'en avaient pas. MEDIA et M2M ont fourni des jeux de développement. Pour ATIS, nous avons constitué le jeu de développement avec les 500 derniers énoncés du jeu d'entraînement originel car les énoncés ne sont pas rangés par intention. Pour SNIPS et SNIPS70, les jeux de développement que nous avons constitués sont composés des 500 derniers énoncés de leurs jeux d'entraînement respectifs après mélange (les énoncés étaient rangés).

4 Génération

Le but de cet article est de faire une analyse systématique d'une méthode de génération. Pour cela il nous faut une méthode de génération à analyser.

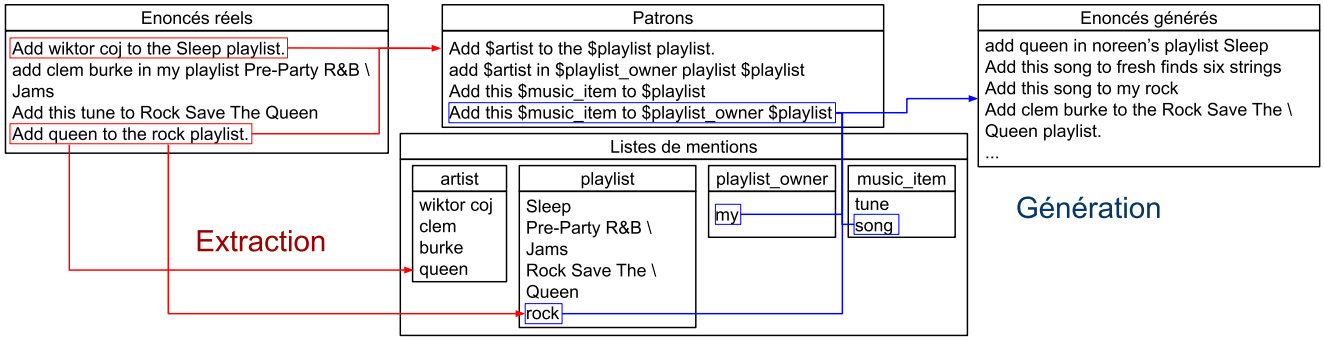

FIGURE 1 – Processus d'extraction et de génération. En rouge, l'extraction effectuée à partir des énoncés de référence afin de constituer nos données brutes pour la génération. En bleu, le processus de génération combine les données brutes afin de créer de nouveaux énoncés. Notre objectif est d'étudier les performances du modèle obtenu à partir des énoncés générés à partir des données brutes extraites des énoncés du benchmark réel.

4.1 Méthode de Génération

La méthode de génération évaluée est une méthode simple de remplissage de patrons, comme le montrent la figure 1 et le tableau 3. Les patrons sont des énoncés dont les mentions ont été remplacées

par leurs classes ou concepts. La méthode de remplissage de patrons consiste à remplir les concepts trouvés dans un patron, $C_{pattern}$, avec les mentions des concepts respectifs, m_c. Cela donne une quantité d'énoncés possibles, p, qui explose très rapidement :

$$p_{pattern} = \prod_{c \in C_{pattern}} Card(m_c)$$

Comme il n'est pas possible de générer tous les exemples pour toutes les quantités de données, le nombre d'énoncés générés a été choisi pour être de $n = 20.000$ pour toutes les différentes quantités de données. Chaque modèle est utilisé pour générer des énoncés $q = n/\#patterns$ par échantillonnage sans remplacement. Cette étape est simplifiée en faisant un échantillonnage des indices se trouvant dans $[\![0, p_{pattern}]\!]$ et en faisant l'extraction des indices des mentions dans leur liste grâce à une série de divisions euclidiennes. Si le modèle ne peut pas générer suffisamment d'énoncés, les énoncés générés sont répétés jusqu'à ce que la quantité requise soit atteinte.

Énoncé	Is	there	a	flight	to	Atlanta
Étiquette	O	O	O	O	O	B-city
Patron	Is	there	a	flight	to	$city
Énoncé généré	Is	there	a	flight	to	Paris

TABLE 3 – Exemple du procédé d'extraction et de génération. Le patron ($3^{ème}$ ligne) est extrait d'un énoncé étiqueté ($1^{ère}$ et $2^{ème}$ lignes). Ce patron est rempli par des mentions ayant le même concept ($city est remplacé par Paris) pour former un nouvel énoncé ($4^{ème}$ ligne). Dans cet exemple, la longueur de la mention est la même, mais en pratique elle peut changer (par exemple : $city peut être rempli par New York).

4.2 Evaluation de la méthode de génération

L'évaluation de la méthode de génération se fait à travers les performances d'un modèle appris sur les données générées, puis validé sur les données de développement et testé sur les données de test. La comparaison entre les performances de ces modèles et celles des modèles entraînés sur les données réelles donne une indication des performances de la génération. Pour que cette comparaison soit pertinente, les modèles et les mentions utilisés pour la génération sont extraits des données réelles, comme le montre la figure 1.

5 Expériences et résultats

Dans cette section, nous décrivons les expériences et les conclusions qui peuvent être tirés des résultats.

5.1 Préparation des données

Nous voulons tester les performances d'un modèle sur plusieurs quantités de données pour connaître l'intérêt de la méthode de génération en fonction des données disponibles. Pour cela nous préparons

les données réelles servant de données brutes. Nous effectuons un découpage en parties de taille croissante contenant les parties plus petites. Cela nous garantit qu'il y a de plus en plus d'information dans les jeux de données. Les données brute servent ensuite à construire les jeux d'entraînement comme décrit Figure 2. Les modèles sont entraînés sur ces données et produisent les résultats obtenus sur les figures 3 et 4 sur les données de test. Nous avons testé différentes manières de distribuer les données au cours de leur génération. Les changements dans les distributions ont été effectués en modifiant la manière dont les patrons et les mentions étaient distribués dans leurs listes respectives, comme le montre la figure 1, puisque notre méthode de génération distribue uniformément les mentions trouvées dans les listes au sein des patrons.

Les multiples distributions testées sont :

Uniforme Les patrons et mentions sont distribué uniformément(c'est à dire qu'il n'y a qu'une apparition dans une liste d'un patron ou d'une mention).

Distribution réelle des mentions Les mentions sont distribuées avec leur fréquences réelle d'apparition (c'est à dire que les listes de mentions contiennent le nombre d'apparition d'une mention). Les patrons sont distribués uniformément.

Distribution réelle de patrons Les patrons sont distribués avec leur fréquences réelle d'apparition et les mentions sont ditribuée uniformément.

Distributions réelles Les patrons et les mentions sont distribuées avec leurs fréquences réelles.

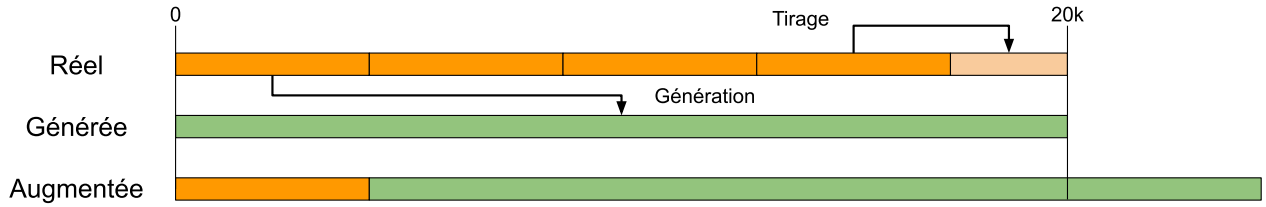

FIGURE 2 – Préparation des données. Les briques oranges symbolisent les données réelles dans leur taille d'origine. Les briques vertes symbolisent les 20.000 générés. Sur la première ligne est représentée la façon dont les données réelles ont été préparées pour l'entraînement. L'échantillonnage est effectué à chaque époque afin de ne pas perturber la distribution sur la totalité de l'entraînement. Sur les deuxième et troisième lignes sont représentées les façons dont les données générées et augmentées ont été préparées.

Pour pouvoir comparer les ensembles générés avec les données réelles utilisées pour les générer, les ensembles de données réels ont été multipliés en longueur afin d'atteindre des énoncés de 20.000, comme le montre la figure 2.

L'augmentation a également été évalué après la génération. Dans ce contexte, la méthode de génération *Distribution réelle des mentions* et la méthode de génération *Distributions réelles* ont été utilisées pour l'augmentation car elles ont donné de meilleurs résultats que les autres méthodes (voir figure 3).

Modèle	ATIS	M2M	SNIPS	SNIPS70	MEDIA
BiLSTM	95.3	91.3	91.8	76.8	84.6
Référence	93.9	92.5	91.8	74.1	85.6

TABLE 4 – Performances du modèle obtenu sur les datasets complets. Le BiLSTM est notre modèle. Le modèle référence est le BiGRU + CRF de (Béchet & Raymond, 2019).

5.2 Système

Le modèle entraîné est un BiLSTM à deux couches [3] (Hochreiter & Schmidhuber, 1997; Greff *et al.*, 2016) de taille caché 128 est ensuite entraîné sur les données. La couche de plongement lexicaux est initialisée avec des vecteurs provenant de Word2Vec (Mikolov *et al.*, 2013) entraîné sur les données d'entraînement. L'ensemble de développement réel est utilisé pour chaque entraînement. La sélection du modèle a été faite en prenant le score de F-mesure le plus élevé obtenu sur l'ensemble de développement. Le score de F-mesure est ensuite calculé sur l'ensemble de test. Le tableau 4 présente l'état de l'art pour les tâches évaluées et les résultats obtenus par notre modèle entraîné sur les données d'entraînement réelles. D'une manière générale, les résultats sont comparables, ce qui rendra nos analyses applicables à tout type de systèmes similaires.

6 Résultats

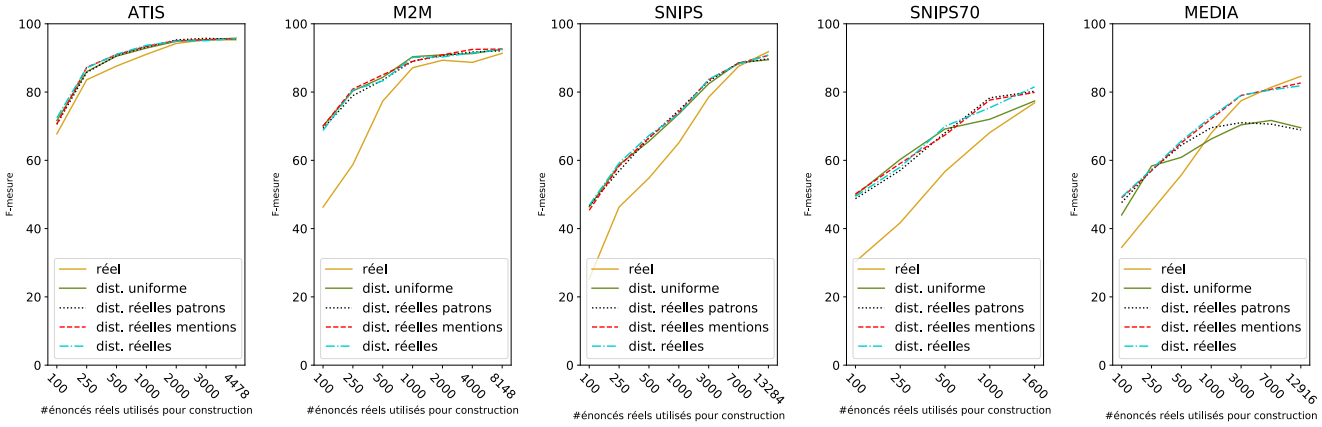

FIGURE 3 – Score de F-mesure obtenus avec conlleval, en mode *génération seule*. Les scores sont obtenus sur le jeu de test à partir de modèles entraînés sur les données préparées à partir du nombre d'énoncés réels trouvés en abscisse.

Le score F-mesure a été mesuré sur l'ensemble des tests de chaque corpus. En mode génération seule, d'après les résultats présentés dans la figure 3, la génération améliore les performances du modèle pour de petites quantités de données, qu'importe la méthode de distribution utilisée. À mesure que les données initiales sont de plus en plus grandes, l'écart entre les modèles entraînés sur les données générées uniformément et les données réelles est réduit, à l'exception de la tâche MEDIA où les modèles entraînés sur les données générées uniformément perdent en performance. Pour la tâche

3. Hyperparamètres : optimiseur Adam avec un taux d'apprentissage initial de 0,001 , plongement lexicaux initialisé avec Word2Vec de taille 300 sur les données d'entraînement, CrossEntropy loss.

SNIPS70, il n'y a pas assez de données disponibles pour atteindre le point où l'écart se referme mais la même tendance peut être observée. En ce qui concerne le mode *augmentation de données* (voir Table 4), on constate que dans l'ensemble les résultats obtenus sont légèrement meilleurs que les résultats dans le mode *génération de données*, notamment sur une grande quantité de données.

Les données générées suivant la distribution *Distribution réelle des patrons* ont tendance à être moins efficaces que les autres jeux de données générés suivant une distribution réelle, en particulier sur la tâche MEDIA où les performances sur de grandes quantités de données sont les pires. En général, les sytèmes appris sur les données générées suivant les *Distribution réelle de mentions* et les *Distributions réelles* tendent à donner de meilleurs résultats mais ont toujours des résultats inférieurs pour SNIPS et MEDIA sur les données générées à partir de beaucoup de données brutes.

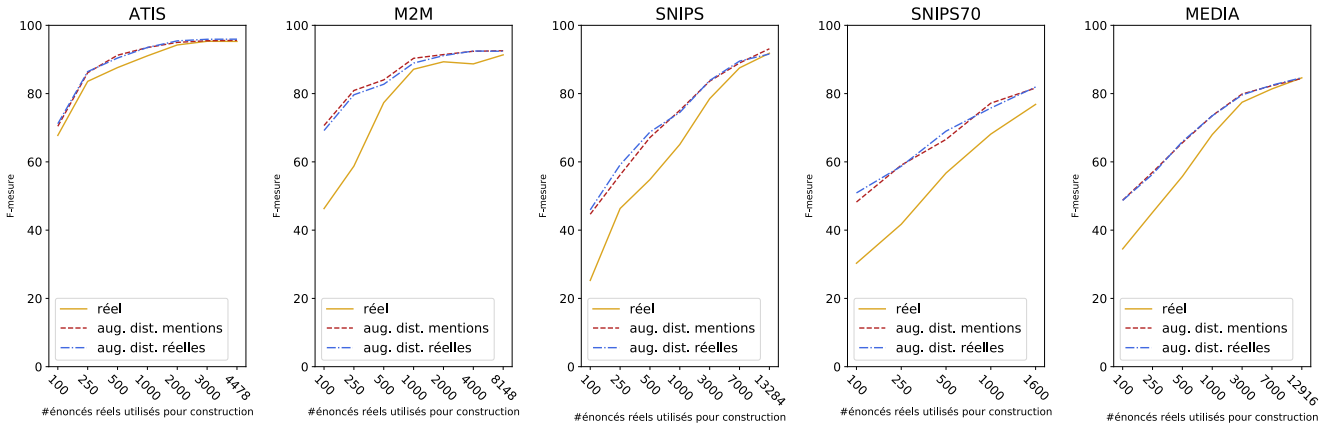

FIGURE 4 – Score de F-mesure des données augmentées calculés avec conlleval, en mode *augmentation de données*. Les scores sont obtenus sur le jeu de test à partir de modèles entraînés sur les données préparées à partir du nombre d'énoncés réels trouvés en abscisse.

7 Analyse des données

Les résultats obtenus à partir des ensembles générés sont bons pour la plupart des tâches. Ce n'est pas le cas pour la tâche MEDIA. Le but de cette section est de trouver les explications de ces performances inférieures à partir d'une série d'analyse de données. Ces analyses visent à découvrir comment le processus de génération perturbe la manière dont les données sont distribuées et comment les distributions initiales des données de certaines tâches peuvent induire leur niveau de difficulté. Un deuxième objectif est de conclure sur de bonnes pratiques concernant la manière dont les données devraient être distribuées pour obtenir de meilleures performances, avec et sans génération.

Comme le montre la figure 3, la modification de la distribution uniforme des mentions en faveur de la distribution réelle a réglé une grande partie des problèmes de performance sur MEDIA. Une analyse plus approfondie est nécessaire pour comprendre la véritable nature de ce changement et pour savoir s'il est possible ou non de trouver des règles générales qui pourraient s'appliquer aux distributions de mentions.

7.1 Distribution des mentions

La distribution des mentions semble avoir une grande importance d'après la figure 3. Ne pas suivre la vraie distribution peut créer des problèmes de performance comme les résultats de MEDIA le montrent. Y a-t-il des indicateurs dans les données de la différence entre les jeux de données avec des mentions uniformément distribuées et les autres jeux de données ? Le premier et le plus simple des indicateurs est la longueur de la mention en nombre de tokens.

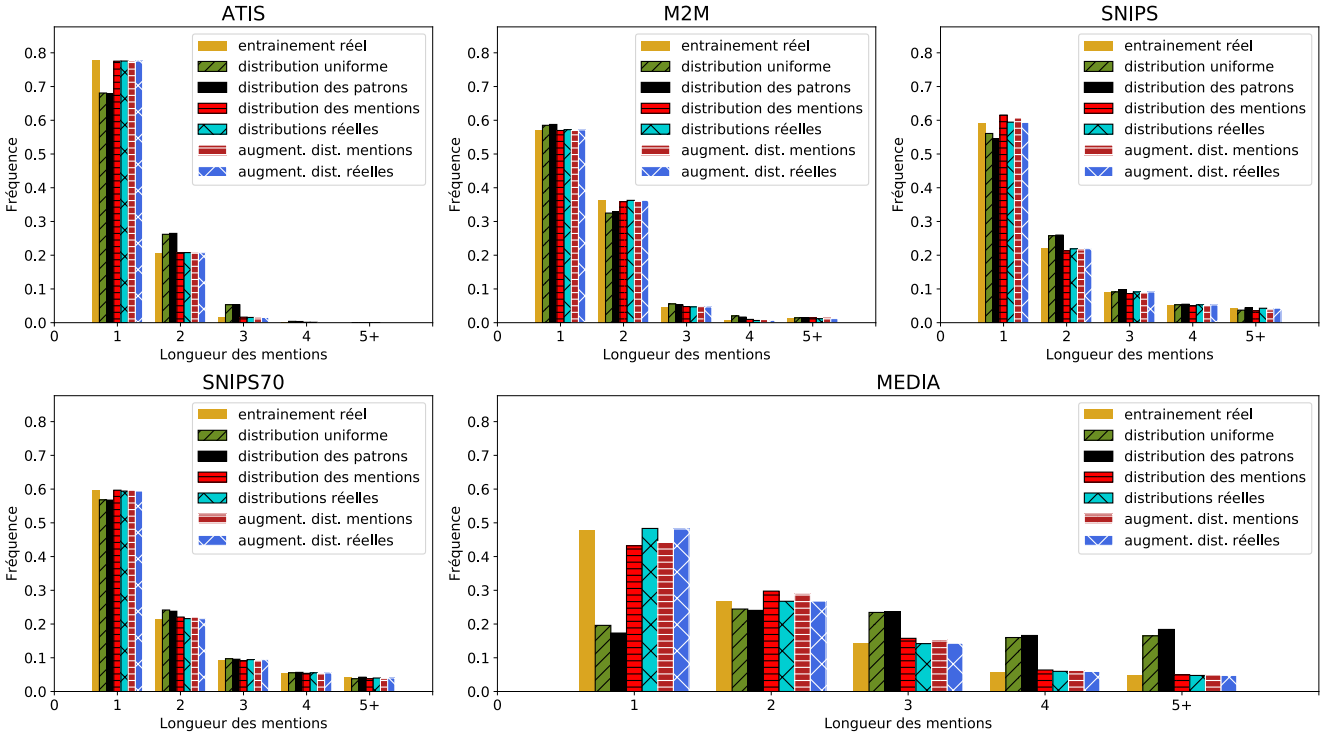

FIGURE 5 – Longueur des mentions en nombre de tokens.

La distribution des mentions basée sur la longueur des mentions, comme le montre la figure 5, montre une tendance commune entre les jeux de données à avoir leurs mentions distribuées de manière similaire à une distribution Zipfienne en fonction de leur nombre de tokens. Pour la plupart des ensembles de données, cette distribution n'est que faiblement influencée par l'échantillonnage uniforme des mentions. Cependant, pour MEDIA, la distribution est complètement faussée comme le montre la figure 5. Cela montre que dans les données réelles de MEDIA, les mentions avec un nombre de tokens inférieur apparaissent avec une plus grande multiplicité que les mentions avec un nombre de tokens supérieur. Ce problème peut être facilement résolu lorsque les mentions sont extraites avec leur multiplicité, mais dans un cas d'une utilisation où il n'y a pas d'énoncés à partir desquels extraire des informations de distribution, cela pourrait constituer un obstacle.

7.2 Impact des patrons

La distribution des patrons ne semble pas avoir d'impact positif sur les performances d'après les résultats Figure 3. Toutefois, cela ne signifie pas que la quantité de patrons n'est pas pertinente pour les performances. L'expérience conçue pour étudier l'impact de la quantité de modèles consiste à étudier l'évolution du score de F-mesure en fonction de la quantité de patrons utilisés. Les listes de mentions

ont été réduites de moitié afin d'avoir un nombre suffisant de mentions pour que le modèle puisse apprendre avec de la variabilité mais leur donner un impact plus faible sur le score. Le pourcentage de mentions de l'ensemble de test trouvé dans l'ensemble d'entraînement, ou recouvrement, a également été calculé afin d'avoir une idée de l'impact des patrons sur les représentations des mentions et de leurs classes.

Les résultats de cette expérience ont montré Figure 6 que le recouvrement des mentions tend à atteindre son plateau final à environ 100 patrons. Cependant, l'augmentation des performances des modèles a tendance à ralentir après 250 à 500 patrons. Cela montre qu'il faut une certaine variété dans le contexte des mentions pour atteindre de meilleures performances.

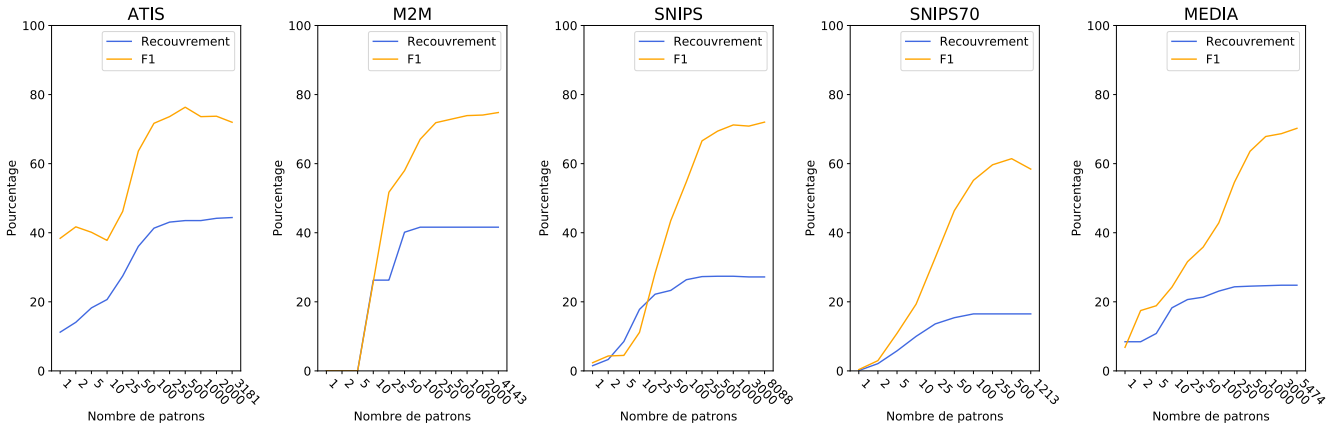

FIGURE 6 – F-mesure dépendant du nombre de patrons. Le recouvrement est le pourcentage de mentions du jeu de test trouvé dans le jeu d'entraînement généré. Cette expérience à été réalisée avec la moitiée des mentions. Les cinq premiers patrons tirés de M2M se trouvent être des patrons vides(ne contienne pas de mentions), ce qui explique les résultats nuls.

7.3 Ambiguïté

Un problème courant dans le traitement du langage naturel est l'ambiguïté. Nous avons choisi de mesurer l'ambiguïté parce qu'elle peut influencer la difficulté d'un jeu de données. Nous mesurons l'ambiguïté en comptant le nombre de concepts dont un token fait partie. Les extérieurs (étiquette O) ne sont pas considérés comme des concepts. L'ambiguïté est faible pour M2M car près de 75% des tokens n'ont pas de concept. SNIPS est peu ambigu, mais sur un nombre élevé de tokens, près de 80% d'entre eux ont un concept, mais peuvent également être extérieurs. ATIS et MEDIA sont ambigus sur une plus petite proportion de leurs tokens, mais avec un nombre plus élevé de concepts par token dans l'ensemble. L'ambiguïté telle que nous la mesurons ne semble pas avoir un rôle significatif dans l'évaluation de la difficulté de la tâche car elle est difficilement comparable pour tous les ensembles de données.

8 Analyses des résultats

Dans cette section, nous compilons et tirons des conclusions sur les résultats et les analyses effectuées. Ces conclusions prennent la forme de lignes directrices sur la manière d'utiliser la méthode de

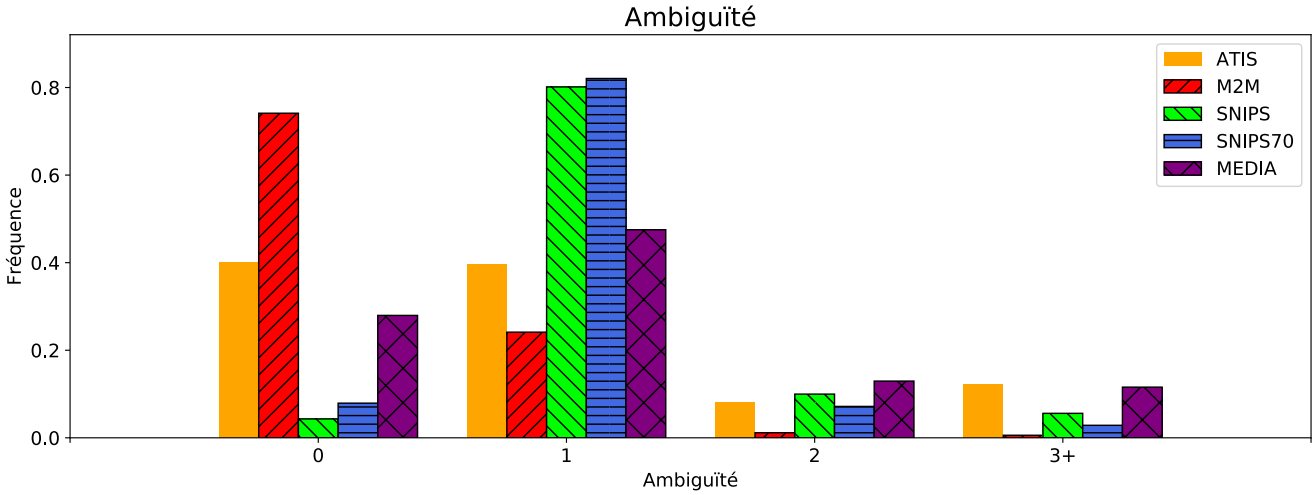

FIGURE 7 – L'ambiguité mesuré comme le nombre de concepts auquel peut appartenir un token. 0 est quand un token ne peut être qu'extérieur. n est quand un token fait partie de n concepts, mais peut aussi être un extérieur. L'ambiguité montré est l'ambiguité du jeu de données réelle de chaque corpus, car l'ambiguité ne varie quasiment pas avec la génération.

génération. Ces lignes directrices sont valables lorsque le modèle d'étiquetage utilisé est un BiLSTM, des recherches supplémentaires sont nécessaires pour pouvoir affirmer que ces lignes directrices réagissent de manière similaire avec d'autres types de modèles (par exemple les CRF).

8.1 Génération

L'utilisation la plus élémentaire d'une méthode de génération est de l'utiliser uniquement pour la génération. Dans cette configuration, nous avons vu qu'il y a quelques points qui doivent être respectés pour générer efficacement. Tout d'abord, nous avons vu que la diversité des patrons est très importante jusqu'à un certain point, après quoi la diminution du rendement peut rendre prohibitif la recherche de nouveaux patrons. Ce point se situe entre 250 et 500 patrons. Le deuxième point principal qui semble ressortir de l'analyse des données est que les mentions doivent être distribuées avec une distribution de type Zipfienne basée sur la longueur des mentions. Le respect de ces points devrait permettre d'obtenir un système aux performances correctes.

8.2 Augmentation

L'utilisation la plus courante de la méthode de génération indiquée ci-dessus est très probablement l'augmentation. Pour de faibles quantités de données, d'après nos observations, l'augmentation et la génération ont tendance à avoir des performances similaires. Il s'agit probablement d'un biais dans la façon dont les données générées et les données réelles sont distribuées dans les ensembles de données augmentées. Si on dispose de données réelles, il est possible d'utiliser les distributions des données réelles pour la génération ou l'augmentation. Cela améliore globalement les résultats. La meilleure utilisation de l'augmentation se situe au niveau des grandes quantités de données où elle tend à donner de meilleurs résultats que les données réelles ou générées, ce qui remet en question la répartition des données notre augmentation.

8.3 Classement des Corpus

Les analyses effectuées sont complémentaires à ce qui est décrit dans l'article (Béchet & Raymond, 2019). Leur travail est davantage axé sur les modèles et les performances, contrairement à notre approche, qui est davantage axée sur le contenu des corpus. Les critères que nous utilisons pour classer les tâches sont : la performance de base de la méthode de génération, la longueur des mentions et le taux d'ambiguïté. Dans l'ensemble, le classement reste le même que celui présenté dans (Béchet & Raymond, 2019) : M2M > ATIS > SNIPS > SNIPS70 > MEDIA.

9 Conclusions et travaux futurs

Nous avons constaté et validé au cours de ce travail que la méthode de génération par remplissage de patrons est utile pour de faibles quantités de données. Cependant, il n'y a que peu d'amélioration significative des performances pour des quantités plus importantes de données si cette méthode est utilisée sans aucune information sur la distribution. Les performances peuvent même se dégrader comme le montre MEDIA.

Ces problèmes nous ont poussés à tester d'autres façons d'utiliser la méthode de génération, comme le test avec les distributions de mention réelle ou l'augmentation des données. Cela nous a également poussés à faire une analyse des corpus afin d'essayer de comprendre pourquoi de tels problèmes pouvaient survenir. Nous avons pu constater que les mentions et la manière dont elles sont distribuées dans un ensemble de données est un facteur clé dans les performances obtenues à partir de cet ensemble de données. C'était le principal facteur expliquant pourquoi les performances de MEDIA étaient aussi dégradées qu'elles l'étaient, et en réglant ce problème avec la distribution réelle, nous avons montré que nous pouvons obtenir des performances proches de la réalité avec les données générées. L'analyse que nous avons effectuée confirme également la classification présentée (Béchet & Raymond, 2019).

Nous ne pouvons pas conclure sur une méthode d'utilisation de la génération, mais nous pouvons mettre en évidence les meilleures pratiques. Nos résultats dépendent des modèles que nous avons utilisés pour l'évaluation des performances (ici, un BiLSTM), et les expériences nécessiteraient d'être réalisées avec d'autres types de modèles (par exemple des CRF). Ces meilleures pratiques consistent à mettre l'accent sur la distribution de mentions plus courtes et à disposer d'un nombre raisonnable de patrons pour avoir un contexte varié.

Dans les travaux futurs, nous aborderons les points non résolus tels que la recherche et l'évaluation de méthodes d'estimation des distributions de mentions. Avec les résultats sur l'impact de la variété des patrons, nous travaillerons également sur la conception et l'essai de méthodes de génération ou d'augmentation des patrons. L'augmentation doit également être étudiée plus en détail, en particulier pour les faibles quantités de données pour lesquelles notre préparation actuelle des données pourrait limiter les performances de l'augmentation aux performances du dispositif de génération. En voyant comment la singularité de MEDIA nous a donné des indices pour trouver de meilleures méthodes de génération, les travaux futurs porteront sur d'autres corpus connus d'étiquetage de phrases. Enfin, notre travail ne concernait que la génération des données d'entraînement, mais pour qu'une méthode de génération puisse être utilisée, il est nécessaire d'étudier comment générer au mieux les données d'entraînement et de validation.

Références

BÉCHET F. & RAYMOND C. (2019). Benchmarking Benchmarks : Introducing New Automatic Indicators for Benchmarking Spoken Language Understanding Corpora. In G. KUBIN & Z. KACIC, Éds., *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*, p. 4145–4149 : ISCA.

BONNEAU-MAYNARD H., ROSSET S., AYACHE C., KUHN A. & MOSTEFA D. (2005). Semantic annotation of the french media dialog corpus. In *Ninth European Conference on Speech Communication and Technology*.

COUCKE A., SAADE A., BALL A., BLUCHE T., CAULIER A., LEROY D., DOUMOURO C., GISSELBRECHT T., CALTAGIRONE F., LAVRIL T. *et al.* (2018). Snips voice platform : an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv :1805.10190*.

DAHL D. A., 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. In *Proceedings of the workshop on Human Language Technology*, p. 43–48 : Association for Computational Linguistics.

GREFF K., SRIVASTAVA R. K., KOUTNÍK J., STEUNEBRINK B. R. & SCHMIDHUBER J. (2016). LSTM : A search space odyssey. *IEEE transactions on neural networks and learning systems*, **28**(10), 2222–2232.

HOCHREITER S. & SCHMIDHUBER J. (1997). Long short-term memory. *Neural computation*, **9**(8), 1735–1780.

KAFLE K., YOUSEFHUSSIEN M. & KANAN C. (2017). Data augmentation for visual question answering. In *Proceedings of the 10th International Conference on Natural Language Generation*, p. 198–202.

KINGMA D. P. & WELLING M. (2014). Auto-Encoding Variational Bayes.

MIKOLOV T., SUTSKEVER I., CHEN K., CORRADO G. S. & DEAN J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, p. 3111–3119.

NEURAZ A., LLANOS L. C., BURGUN A. & ROSSET S. (2018). Natural language understanding for task oriented dialog in the biomedical domain in a low resources context. *arXiv preprint arXiv :1811.09417*.

PAN S. J. & YANG Q. (2010). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, **22**(10), 1345–1359.

RUDER S. (2019). *Neural Transfer Learning for Natural Language Processing*. Thèse de doctorat, National University of Ireland, Galway.

SHAH P., HAKKANI-TÜR D., TÜR G., RASTOGI A., BAPNA A., NAYAK N. & HECK L. (2018). Building a conversational agent overnight with dialogue self-play. *arXiv preprint arXiv :1801.04871*.

WESTON J., BORDES A., CHOPRA S. & MIKOLOV T. (2016). Towards ai-complete question answering : A set of prerequisite toy tasks. In Y. BENGIO & Y. LECUN, Éds., *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

WILLIAMS J. D., ASADI K. & ZWEIG G. (2017). Hybrid code networks : practical and efficient end-to-end dialog control with supervised and reinforcement learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 665–677, Vancouver, Canada : Association for Computational Linguistics. DOI : 10.18653/v1/P17-1062.

YOO K. M., SHIN Y. & LEE S.-G. (2019). Data augmentation for spoken language understanding via joint variational generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, p. 7402–7409.