
Institut National des Langues et Civilisations Orientales

Département Textes, Informatique, Multilinguisme

Comparaison de systèmes de génération automatisée de textes

Génération de textes immobiliers avec des systèmes de GAT avec modèles et
réseaux neuronaux

MASTER

TRAITEMENT AUTOMATIQUE DES LANGUES

Parcours :

Ingénierie Multilingue

par

Maëva LEPROUX

Directeur de mémoire :

Mathieu Valette

Encadrant entreprise :

Eurielle Pignard

Année universitaire 2019/2020

TABLE DES MATIÈRES

Liste des figures	5
Liste des tableaux	5
Introduction	7
I Contexte général	13
1 État de l'art	15
1.1 Introduction	15
1.2 Périmètre de recherche	16
1.3 Méthodes	18
1.4 Evaluations	19
1.5 Conclusion	22
2 Description du projet	23
2.1 Introduction	23
2.2 Contexte industriel	23
2.3 Discussion	26
2.4 Conclusion	27
II Expérimentations	29
3 Méthodes	31
3.1 Introduction	31
3.2 Mise en place d'un système de GAT avec modèles	31
3.3 Mise en place d'un système de GAT avec Deep Learning	41
3.4 Conclusion	43
4 Résultats	45
4.1 Introduction	45
4.2 Système de GAT avec modèles	45
4.3 Système de GAT avec réseau neuronal	47
4.4 Conclusion	51
5 Evaluation	53
5.1 Introduction	53
5.2 Type d'évaluation	53
5.3 Analyse des résultats des évaluations	56
5.4 Que retenir de cette évaluation?	59

5.5 Conclusion	59
Conclusion générale	61
Bibliographie	63
III Annexes	67
A Documentation	69
A.1 Langages et bibliothèques	69
A.2 Corpus	69
A.3 Evaluation	71
B Extraits de scripts	75
B.1 Script contenant la configuration du réseau de neurones	75
B.2 Script pour l'entraînement du modèle	76
B.3 Script permettant de lancer la génération des textes	77
Index	79

LISTE DES FIGURES

2.1	Principales étapes d'un projet avec Scribt	25
4.1	Exemple de texte généré par le système de GAT avec modèles	47
5.1	Formulaire d'évaluation fourni aux évaluateurs pour chacun des textes . .	55
A.1	Formulaire d'évaluation fourni aux évaluateurs pour chacun des textes . .	73
B.1	Configuration du système (<i>config.py</i>)	75
B.2	Script d'entraînement du système (<i>train.py</i>)	76
B.3	Script de génération de textes (<i>generate.py</i>)	77

LISTE DES TABLEAUX

1.1	Exemple de résultats du système BabyTalk de [Kulkarni et al., 2013]	17
1.2	Tableau illustrant l'exemple de [Gatt and Krahmer, 2018] sur les modèles (template)	19
1.3	Hallucination dans les systèmes de GAT neuronaux [Dušek et al., 2018] . .	20
3.1	Texte cible du projet "Evolution des prix de l'immobilier"	32
3.2	Texte cible du projet "Evolution des prix de l'immobilier" avec les données mises en valeur	33
3.3	Exemple de modèles pour une partie du texte cible	34
3.4	Exemple de création de sous-parties	35
3.5	Exemple d'insertion de balises dans les modèles réalisés pour une des parties	35
3.6	Exemple de phrase avec les différentes balises	36
3.7	Exemple de tableau regroupant les informations sur les données d'entrée requis	37
3.8	Exemple de données pour les catégories présentées dans le tableau précédent	38
3.9	Exemple de donnée de taux avec ou sans signes	39
3.10	Exemple de résultats souhaités pour le cas du gentilé	40
3.11	Exemple de sous-parties créées pour la phrase introductive	40
4.1	Exemple de document de test pour le projet "Evolution des prix de l'immo- bilier"	46
4.2	Catégorisation des erreurs dans les textes générés via modèle	46
4.3	Exemple de phrases générées via textgenrnn avec les paramètres par dé- faut, avec un entraînement de 50 epoch et un corpus d'un texte	48
4.4	Exemple de phrases générées via textgenrnn avec les paramètres par dé- faut, avec un entraînement de 50 epoch et un corpus de 10 textes	48
4.5	Exemple de phrases générées via textgenrnn avec les paramètres par dé- faut, avec un entraînement de 100 epoch et un corpus de 30 textes	49
4.6	Exemple de phrases générées via textgenrnn avec les paramètres par dé- faut, avec un entraînement de 200 epoch et un corpus de 30 textes	49

4.7	Exemple de phrases générées via textgenrnn avec des paramètres définis, avec un entraînement de 100 epoch et un corpus de 30 textes	50
4.8	Exemple de phrases générées via textgenrnn avec des paramètres définis, avec un entraînement de 200 epoch et un corpus de 30 textes	50
5.1	Moyenne des résultats de l'évaluation sur les 10 textes	57
5.2	Médiane des résultats de l'évaluation sur les 10 textes	57
5.3	Moyenne et médiane des notes attribuées aux trois critères pour l'ensemble des textes	58
5.4	Moyenne et médiane des notes attribuées aux textes 1 à 5	58
5.5	Moyenne et médiane des notes attribuées aux textes 6 à 10	58
A.1	Langages et logiciels utilisés dans le cadre de ce mémoire	69
A.2	Extrait d'un des corpus utilisé pour la génération avec réseau neuronal . .	71
A.3	Extrait des commentaires recueillis lors de l'évaluation	72

Résumé

Ce mémoire décrit la procédure d'un projet de génération de textes automatisée avec deux systèmes différents. Nous avons mis en place un corpus de modèles, de données d'entrée, rédigé les règles prises en compte dans le projet et programmé un système à base de réseau neuronal. Les deux systèmes doivent générer des textes à partir de données immobilières sur des villes et leur département. Ces textes doivent être compréhensibles et factuels, dans un but informatif. Nous comparerons les résultats obtenus par les deux systèmes de génération automatique de texte, ainsi que les notations obtenues pendant leur évaluation humaine. Nous détaillerons ainsi les spécificités des systèmes, ainsi que les avantages et inconvénients de chacun.

Mots clés : génération automatique de textes, évaluation humaine, système de génération automatique de texte à base de modèles, système de génération automatique de texte à base de réseau neuronal, textes immobiliers

Remerciements

Je tiens tout d'abord à remercier mon directeur de mémoire, Mathieu Valette, pour ses réponses pertinentes et son soutien tout au long de la rédaction de ce mémoire, ainsi que l'équipe pédagogique du Master pour l'enseignement dont nous avons pu profiter pendant ces deux dernières années. Je souhaite ensuite remercier mon encadrante entreprise, Eurielle Pignard, ainsi que mes collègues de stage, et particulièrement Saskia Branger et Alexandra Guérin pour leur aide et leur amitié. Je remercie également Lili Wu, Aphala Phaxay, Clémence Laurent et mes camarades pour leur compagnie durant ces deux années de Master. Je remercie Martial Bencteux pour son soutien sans faille et ses remarques pertinentes, ainsi qu'Amélie Bencteux, Eliane Proust, Michel Proust et Pauline Nallet pour leur aide précieuse. Enfin, je remercie ma mère, qui a toujours cru en moi, ainsi que mon père et mes grands parents.

INTRODUCTION

Présentation générale

Ce mémoire est le résultat d'un projet de comparaison de systèmes de génération automatique de textes, l'un ayant eu lieu en entreprise à LABSENSE, l'autre ayant été testé lors de la rédaction de ce mémoire, le projet s'étant déroulé dans le cadre du master Traitement Automatique des Langues (TAL) de l'Inalco. L'objectif de ce mémoire est d'établir un comparatif de performances et de résultats de ces deux systèmes de GAT testés, et de convenir des avantages et inconvénients de chacun. Dans ce cadre, nous avons ainsi utilisé un système de GAT fonctionnant avec des modèles, pour lequel nous avons construit un corpus de phrases, ainsi que de données d'entrées, et établi des règles précises. Nous avons également utilisé un système de GAT avec réseau neuronal, avec lequel nous avons effectué de nombreux tests pour trouver une configuration permettant d'obtenir des résultats semblables à ceux attendus. Nous avons ainsi comparé nos résultats. Également, nous avons produit une évaluation humaine permettant d'avoir un aperçu plus large de la qualité des textes obtenus.

Problématique de recherche

L'objectif de ce mémoire est d'apporter une réflexion sur ces quelques questions :

- quelles sont les spécificités des différents systèmes de GAT?
- quelles sont les différentes étapes d'un projet de génération de texte automatisée?
- comment faire une évaluation d'un système de GAT?

Nous allons tenter de répondre à ces quelques questions dans ce mémoire.

Plan

Le chapitre 1 contient un état de l'art de la génération automatique de textes qui introduit les avancées et particularités du milieu de la recherche concernant les systèmes de GAT. Le chapitre 2 est une description du projet et une réflexion sur la problématique de recherche abordée dans ce mémoire. Dans le chapitre 3, nous décrivons les méthodes employées pour produire les projets de génération automatisée de texte, ainsi que les corpus utilisés. Dans le chapitre 4, nous présentons les textes générés. Dans le chapitre 5, nous présentons une évaluation humaine de nos textes obtenus.

GLOSSAIRE

BLEU : Bilingual Evaluation Understudy

CSV : Comma-separated values

GAT : Génération Automatique de Texte

NLG : Natural Language Generation

NLU : Natural Language Understanding

ROUGE : Recall-Oriented Understudy for Gisting Evaluation

SEO : Search Engine Optimization

TAL : Traitement Automatique de la Langue

Première partie
Contexte général

ÉTAT DE L'ART

Sommaire

1.1	Introduction	15
1.2	Périmètre de recherche	16
1.2.1	Domaine d'études	16
1.2.2	Domaines concernés	16
1.2.3	Données en entrée	17
1.3	Méthodes	18
1.3.1	Systèmes de GAT	18
1.3.2	Modèles/approches	19
1.4	Evaluations	19
1.4.1	Que mesure-t-on?	20
1.4.2	Evaluation automatique et évaluation humaine	21
1.5	Conclusion	22

1.1 Introduction

Dans le domaine du traitement automatique de la langue, la génération automatique de textes s'est créée une place et est devenue une discipline à part entière. De nombreuses applications citées par [Gatt and Krahmer, 2018] ont été développées au fil des années, qui peuvent générer du texte à partir de textes déjà existants, comme des résumés de phrases ou de textes pour les rendre plus concis [Clarke and Lapata, 2010] ou la génération automatique d'évaluation par les pairs pour des articles scientifiques [Bartoli et al., 2016]. Dans le domaine du TAL, la GAT se distingue par sa polyvalence. En effet, cette discipline peut être adoptée dans de nombreux domaines et représenter un gain de temps et d'efforts considérable. Pour avoir une meilleure idée de la GAT, [Reiter and Dale, 2000] nous en apportent une définition, qui est reprise dans [Gatt and Krahmer, 2018] :

Le sous-domaine de l'intelligence artificielle et de la linguistique informatique qui concerne la construction de systèmes informatiques capables de produire des textes compréhensibles en anglais ou dans d'autres langues à partir de représentation de données non linguistiques sous-jacentes.¹

1. the subfield of artificial intelligence and computational linguistics that is concerned with the construction of computer systems that can produce understandable texts in English or other human languages from some underlying non-linguistic representation of information.

Nous pouvons donc définir plusieurs éléments à approfondir ci-dessous, à savoir les différents systèmes de la GAT et leur structure, le contexte et les applications concernés, et l'évaluation possible.

1.2 Périmètre de recherche

1.2.1 Domaine d'études

D'après [Yang et al., 2004], [Yngve, 1961] est un des premiers systèmes de génération, en effet il s'agit d'un système de génération aléatoire de phrases en anglais sur la base d'une grammaire générative. Ce système a été créé pour faire partie d'un système de traduction automatique. En ce qui concerne les systèmes de dialogue, [Zock and Sabah, 1992] citent ELIZA [Weizenbaum, 1966], un système considéré comme le premier agent conversationnel. Ce programme reproduit le comportement d'un psychothérapeute et répond aux questions de patients. Cette discipline a grandement évolué jusqu'à nos jours, avec des travaux plus récents [Papangelis et al., 2019] qui associent la compréhension du langage naturel (Natural Language Understanding) à la GAT pour produire un système qui aide l'utilisateur à trouver un restaurant en lui posant diverses questions. Cependant, la GAT n'a pas uniquement un but informatif. Certes, la grande majorité des systèmes existent pour aider à informer les utilisateurs comme nous l'avons aperçu avec les systèmes cités précédemment, néanmoins, d'autres applications de la GAT reposent sur l'acte d'aider l'utilisateur à appréhender des textes trop complexes pour les enfants [Siddharthan and MacDonald, 2016] ou simplement présentant un nombre de données important [Clarke and Lapata, 2010]. Aussi, la GAT est un domaine assez semblable à celui des résumés de texte (text summarization). En effet, dans ce domaine également nous pouvons trouver de la génération de nouvelles données textuelles. Si la GAT a avant tout un but informatif, nous pouvons tout de même y trouver des systèmes de génération de textes créatifs et divertissants, notamment avec JAPE, développé par [Binsted et al., 1997] qui est capable de générer des blagues et des calembours. Ce système repose sur des modèles fixes (tels que « Quelle est la différence entre X et Y »), et des créneaux variables, type d'approche que nous abordons plus tard. Il est aussi possible de lire des histoires créées par des systèmes de GAT, comme le montre le système BRUTUS de [Bringsjord and Ferrucci, 1999] qui génère des textes basés sur des schémas d'histoires de contes. La GAT présente donc de nombreuses applications, cependant tous les systèmes présentés reposent sur un but précis et unique de génération. Ainsi, il semble compliqué de pouvoir se servir d'un de ces systèmes pour une autre application que celle pour laquelle il a été créé.

1.2.2 Domaines concernés

De nombreux domaines sont concernés par la GAT, généralement à cause de l'important volume de leurs données, ou de leur complexité. Il est également possible d'avoir recours à la GAT dans le cadre d'une rédaction de nombreux textes dans un délai trop rapide pour des rédacteurs humains. Nous pouvons retrouver des systèmes de GAT dans les domaines suivants :

- la météo avec la génération automatique de bulletins météorologiques [Reiter et al., 2005];

- le domaine médical avec la récapitulation d’informations concernant des patients dans un contexte clinique [Hüske-Kraus, 2003];
- la finance avec la génération automatique de rapports sur le prix d’actions [Nesterenko, 2016].
- le journalisme avec la génération automatique d’articles de journaux [Leppänen et al., 2017]
- le sport avec le système PASS, qui génère automatiquement des rapports de matchs de football différents en fonction des équipes [van der Lee et al., 2017]

Les systèmes de GAT répondent à un besoin dans le domaine concerné. Ils permettent un gain de temps et de travail humain non négligeable. Il pourrait être intéressant de développer ces systèmes dans plus de domaines.

1.2.3 Données en entrée

La grande souplesse des systèmes de GAT en termes de données prises en entrée est une des particularités de ce domaine par rapport à la discipline du traitement automatique des langues. En effet, [Gatt and Krahmer, 2018] évoquent les méthodes de text-to-text generation (données textuelles) ou encore data-to-text generation (données non textuelles) utilisées par diverses applications. Pour la génération de textes simplifiés de textes complexes pour les enfants [Siddharthan and MacDonald, 2016] l’application prendra en entrée des données textuelles, de même que pour la génération de questions à but pédagogique [Brown et al., 2005]. Cependant, pour produire des résumés d’informations sur des patients hospitalisés [Banaee et al., 2013] ou des rapports sur la météo ou sur la finance, les systèmes se pencheront sur des données non textuelles, comme des statistiques. Nous pouvons également citer le projet BabyTalk de [Kulkarni et al., 2013], qui prend des images comme données en entrée, dans le but de générer des textes les décrivant.

Images entrées dans le système	Textes générés
	There are one cow and one sky. The golden cow is by the blue sky
	Here we see one person and one train. The black person is by the train
	This is the picture of two dogs. The first dog is near the second furry dog.

TABLE 1.1 – Exemple de résultats du système BabyTalk de [Kulkarni et al., 2013]

La structure d’un système de GAT sera donc en grande partie influencée par le genre de données que l’on peut trouver en entrée, car il doit y figurer une ou plusieurs étapes de traitement de ces données.

1.3 Méthodes

1.3.1 Systèmes de GAT

[Gatt and Krahmer, 2018] donnent 3 approches possibles :

Nous avons d'un côté les architectures modulaires (modular architectures). Les architectures comme celles-ci impliquent des séparations en sous-tâches, avec des différences significatives entre elles. Voici ces sous-tâches :

- *text planner* (que dire) : combine la sélection du contenu et de la structure du texte ;
- *sentence planner* (comment le dire) : combine le rassemblement des phrases, la lexicalisation et la génération des expressions
- *linguistic realiser* (dire) : génère les phrases finales de manière grammaticalement correcte

Ce genre d'architecture est similaire à l'architecture de résumé de texte (analyse des informations et des textes sources, transformation des informations sélectionnées pour rendre le tout plus fluide, puis synthèse du résumé effectué). Nous avons ensuite les architectures planning (planning architectures), c'est l'architecture la plus traditionnelle, avec des perspectives moins modulables, c'est une architecture plus intégrée. Enfin, nous avons les architectures intégrées ou globales (integrated or global architectures). Il s'agit de l'architecture dominante actuellement, qui repose en grande partie sur l'apprentissage statistique des correspondances entre données d'entrée ou de sortie (non-linguistiques).

[Reiter and Dale, 2000] quand à eux nous présentent six étapes généralement présentes dans les architectures de systèmes de GAT, divisées en 3 parties :

- planification de document (document planning)
 - détermination du contenu (content determination) : décider quelle information insérer dans un texte en construction ;
 - structuration du texte (text structuring) : décider l'ordre dans lequel ces informations sortiront dans le texte ;
- microplanification (microplanning)
 - assemblage de phrases (sentence aggregation) : décider quelle information présenter dans des phrases individuelles ;
 - lexicalisation (lexicalisation) : trouver les bons mots pour exprimer l'information ;
- réalisation de surface (surface realisation)
 - génération d'expression référente (referring expression generation) : trouver les mots et phrases pour identifier le domaine ;
 - réalisation linguistique (linguistic realisation) : combiner ces mots et phrases dans des phrases bien formées ;

Les architectures modulaires et les 6 étapes présentées au-dessus sont très similaires et sont donc les méthodes les plus utilisées dans le domaine de la GAT.

1.3.2 Modèles/approches

Les architectures vues précédemment peuvent être mises en place avec diverses méthodes. Celle qui a été jugée la plus « simple » est celle des modèles (templates). Comme le montrent [Gatt and Krahmer, 2018], ce modèle consiste en deux parties, la phrase en elle-même, et les variables qui la composent.

Phrase avec variables	Phrase avec variables remplies avec les données d'entrée
\$player scored for \$team in the \$minute minute.	Ivan Rakitic scored for Barcelona in the 4th minute.

TABLE 1.2 – Tableau illustrant l'exemple de [Gatt and Krahmer, 2018] sur les modèles (template)

Comme nous pouvons le voir, la première phrase est composée de variables, reconnaissables grâce à une écriture particulière, elles sont ici précédées d'un symbole. Ces variables sont ensuite remplies pendant la génération, avec des données structurées, donnant ainsi la deuxième phrase du tableau.

L'avantage des modèles tel que celui proposé dans le tableau ci-dessus, est un plein contrôle de la qualité des textes cible, et de pouvoir éviter la génération de phrases agrammaticales. En revanche, cela implique d'avoir des données d'entrée structurées et des règles précises, afin d'insérer les données correctement dans les variables. Le travail humain demandé pour la rédaction de ces phrases modèles ainsi que la mise en place des règles strictes est un inconvénient non négligeable également.

Récemment, les approches plus statistiques et stochastiques sont devenues populaires dans le milieu de la GAT, avec notamment des approches à base de génération neuronale. [Langkilde-Geary and Knight, 2002] ont été les précurseurs dans ce domaine, avec leur système HALogen. Ces approches permettraient, contrairement aux modèles, de n'avoir que très peu besoin de travail humain pour générer des textes.

1.4 Evaluations

Il est difficile de comparer les différents systèmes directement, car le domaine de la GAT profite d'une très grande variété d'applications et de systèmes. [Gatt and Krahmer, 2018] donnent 2 principales raisons à cela :

- variabilité de la donnée d'entrée (variable input);
- de multiples données d'entrées possibles (multiple possible input);

Il est également utile de noter que l'évaluation d'un système de GAT est complexe car le système en question est en mesure de produire des textes très variables et sans doute avec une qualité différente. Évaluer un texte de ce système n'équivaut pas à faire l'évaluation du système entier.

Pour certains systèmes fonctionnant avec un apprentissage automatique, la question de l'exactitude peut devenir importante. En effet, nous pouvons rencontrer le problème de l'hallucination dans les systèmes de GAT neuronaux [Reiter, 2018] : les systèmes sont susceptibles de générer des textes qui ne sont pas considérés comme "vrais", car contenant des données qui ne sont pas présentes dans les données d'entrée.

Système	Sortie
TR2	Cotto is a coffee shop located near The Bakers.
SLUG-ALT	Cotto is a coffee shop and is located near The Bakers.
TGEN	Cotto is a coffee shop with a low price range. It is located near The Bakers.
GONG	Cotto is a place near The Bakers.
SHEFF2	Cotto is a pub near The Bakers.

TABLE 1.3 – Hallucination dans les systèmes de GAT neuronaux [Dušek et al., 2018]

Avec simplement les données :

```
name[Cotto], eatType[coffee shop], near[The Bakers]
```

Quelques systèmes ont ajouté des informations, comme le système TGEN qui a ajouté la particularité « with a low price range ». Certes, la phrase obtenue est lisible par un humain, mais elle contient des informations ajoutées, dont nous ne pouvons vérifier la véracité. Dans le cas du système SHEFF2 cependant, nous pouvons constater des informations erronées : l'information « eatType[coffee shop] » a disparu et a été remplacée par « pub ». Encore une fois la phrase est bien lisible, mais elle n'est pas vraie.

1.4.1 Que mesure-t-on ?

[Dale and Mellish, 1998] présentent quelques questions fondamentales sur ce sujet :

- Est-ce que le texte est correct d'un point de vue linguistique et grammatical? Est-il lisible et compréhensible pour un humain?;
- Est-ce que le texte contient bien les données d'entrée?;
- Quelle doit être la sortie?;

Il n'y a pas de critère objectif pour juger si un texte est bon ou pas. Tout ce que l'évaluation peut mesurer, c'est la qualité en termes d'exactitude (accuracy) ou de fluidité (fluency).

D'après [Reiter, 2019], il y a 3 dimensions à prendre en compte quand nous évaluons un texte issu d'un système de GAT.

- Fluidité, Lisibilité (linguistic quality, fluency) : est-ce que le texte est facile à lire et à comprendre?;
- Exactitude (accuracy) : est-ce que le texte dans son ensemble est vrai et provient des données d'entrée?;
- Utilité (utility) : est-ce que le texte peut aider un lecteur à accomplir une tâche, ou est-ce qu'il atteint son objectif communicatif?;

Évidemment, ces critères sont malléables dépendant du contexte dans lequel les textes sont produits : pour des textes destinés à des enfants, la question de la maîtrise linguistique sera plutôt « est-ce lisible par un enfant de 10 ans? ».

[Jones and Galliers, 1995] proposent une distinction méthodologique entre les méthodes d'évaluation extrinsèques et intrinsèques :

- l'évaluation extrinsèque mesure la performance du système évalué par rapport à son objectif global ;
- l'évaluation intrinsèque, dans le cas de la GAT, mesure au contraire la performance d'un système via un critère qui ne fait pas référence à d'autres aspects du système, comme par exemple son objectif final vis-à-vis des utilisateurs ;

Pour ces deux méthodes, [Gatt and Krahmer, 2018] donnent des exemples pertinents :

- pour l'évaluation extrinsèque, le changement de comportement effectif suite à des lettres personnalisées incitant le lecteur à arrêter de fumer avec le système STOP [Reiter et al., 2003] est un critère efficace ;
- pour l'évaluation intrinsèque, les éléments recherchés sont les mêmes que ceux cités par [Reiter, 2019] précédemment, à savoir la fluidité, l'exactitude et l'utilité : nous allons rechercher si le texte est qualitatif linguistiquement parlant, et vérifier si les informations des données d'entrée figurent bien dans les textes ;

1.4.2 Evaluation automatique et évaluation humaine

Les systèmes de GAT étaient traditionnellement soumis à une évaluation humaine, d'après [Belz and Reiter, 2006], en montrant aux sujets des textes issus de la GAT et des textes écrits par des humains. Cette méthodologie était utilisée dans les années 90 et est toujours populaire de nos jours. Ces dernières années cependant, la pratique qui consiste à évaluer un système de GAT en le comparant à un corpus de textes écrits par des humains est devenue populaire. [Langkilde, 2002] a été la première à utiliser cette pratique d'évaluation basée sur le corpus : il s'agit de produire l'analyse grammaticale d'un corpus (parse), de la donner au système de GAT et d'ensuite comparer les textes générés aux textes du corpus originel. Plus récemment, [Papineni et al., 2002] ont proposé BLEU (Bilingual Evaluation Understudy), une métrique conçue pour évaluer des systèmes de traduction automatique à l'origine, qui peut calculer la précision (accuracy) d'une traduction automatique en comparant sa proportion de n-grammes à celle d'une traduction de référence. Une autre métrique, ROUGE (Recall-Oriented Understudy for Gisting Evaluation) [Lin and Hovy, 2003], spécialisée pour les résumés de textes, mesure également les n-grammes mais surtout le rappel de ceux-ci entre deux textes.

Cependant, ces mesures ne sont pas infaillibles et ne correspondent pas forcément au milieu de la GAT.

1.5 Conclusion

A l'issue de cet état de l'art, il est possible d'établir une problématique concernant la génération automatique de textes. En effet, il est nécessaire de posséder une certaine connaissance concernant le domaine concerné par le système de GAT produit, et ainsi de pouvoir décider de son architecture et de la méthode utilisée en connaissance de cause. Également, le choix du format des données d'entrées est primordial, que ce soient des données linguistiques ou non, il est nécessaire de produire le système en accord avec ces données. Enfin, selon le système choisi, il est nécessaire d'effectuer une évaluation correspondant aux méthodes choisies, pour obtenir des résultats pertinents par rapport à la performance du système. Il faut également tenir compte de l'objectif du projet et des personnes visées par les textes générés.

DESCRIPTION DU PROJET

Sommaire

2.1	Introduction	23
2.2	Contexte industriel	23
2.2.1	Scribt : un outil de GAT	24
2.2.2	Description de projet Scribt	24
2.3	Discussion	26
2.3.1	Définition d'un projet	26
2.3.2	Définition d'une problématique de recherche	27
2.4	Conclusion	27

2.1 Introduction

Dans cette partie, nous allons présenter le contexte industriel de notre mémoire, et en particulier le logiciel Scribt qui est l'un des systèmes de GAT dont nous allons parler dans la partie Expérimentations. Nous allons décrire cet outil, et expliquer notre démarche de recherche.

2.2 Contexte industriel

Nous avons effectué notre stage de fin d'études dans une entreprise nommée LABSENSE. Dans le cadre de ce stage, nous avons pu avoir un aperçu de l'utilisation d'un système de génération de textes automatisée dans un contexte commercial. En effet, fondée en 2011 et rachetée en 2018 par le groupe Groupe Pratique, la start-up Labsense est une entreprise qui a développé une technologie de rédaction automatisée de contenu qui utilise des données structurées dans le but de générer du langage naturel. L'entreprise se spécialise notamment dans la vente de contenu automatisé à visée commerciale, pour divers milieux comme l'e-commerce, l'immobilier et le tourisme. Avec la maîtrise de langues telles que le français, l'anglais, l'espagnol, le néerlandais ou l'italien, Labsense se positionne sur le marché français mais également étranger.

Dans le cadre de notre projet, nous allons surtout nous intéresser à leur application **Scribt**, un logiciel d'intelligence artificielle sémantique capable de rédiger de façon automatisée du contenu en langage naturel.

2.2.1 Scribt : un outil de GAT

Le logiciel Scribt est utilisé par LABSENSE dans le cadre de projets de génération de textes dans un but commercial. Il est composé de plusieurs éléments, que nous allons décrire dans cette partie.

Backoffice

Le backoffice est l'outil du logiciel Scribt dans lequel les phrases écrites lors de l'étape de rédaction des modèles sont intégrées. En effet, pour chaque phrase rédigée, nous pouvons insérer des balises de synonymes pour ajouter de la variabilisation aux phrases. Egalement, nous pouvons gérer les accords entre les noms et adjectifs, ou verbes avec cet outil.

Interface administrateur

Le paramétrage des textes et des données d'entrée se fait sur cet outil du logiciel Scribt, l'interface d'administration. Dans cette partie, nous pouvons renseigner les règles que nous avons définies lors de la préparation du projet. Ainsi, nous pouvons mettre en forme les données d'entrée et gérer de manière efficace la fusion des modèles avec celles-ci.

Ces deux outils sont les éléments principaux du système de GAT proposé par Labsense. Diverses applications sont également utilisées en complément :

- **Semrush** et les moteurs de recherche classiques, pour la recherche sémantique obligatoire avant de commencer un projet de génération de textes ;
- **Diverses applications Google**, sur lesquelles les phrases et les règles sont rédigées notamment ;
- **Antidote**, un logiciel de correction orthographique utilisé lors du débogage des textes, pour s'assurer de l'absence de fautes ou de répétitions ;

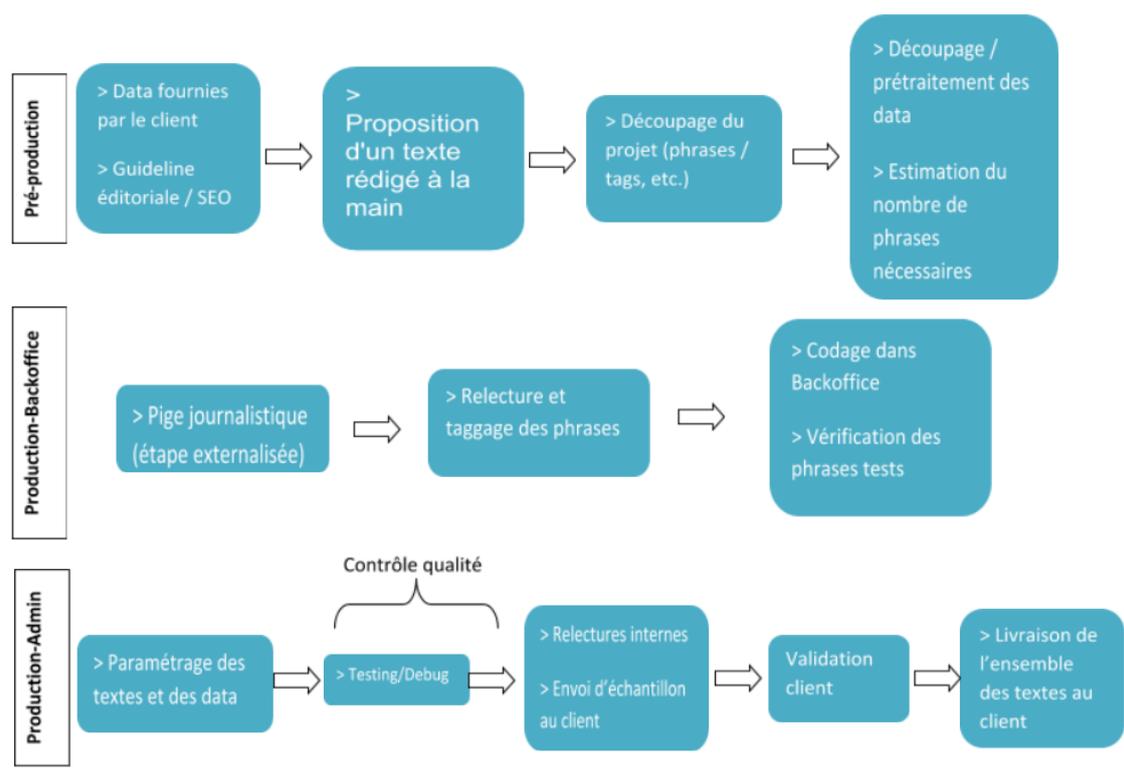
2.2.2 Description de projet Scribt

Avec Scribt, un projet de génération de textes se déroule selon plusieurs étapes précises. Nous allons vous décrire ces différentes étapes en approfondissant le rôle des outils décrits plus haut.

Pré-production

Comme tout projet de GAT, il est nécessaire d'effectuer une recherche sémantique sur le sujet demandé, avec des moteurs de recherche classiques mais également des logiciels comme Semrush. Cette recherche est suivie par la rédaction, à la main, d'un texte contenant les données demandées par le client ainsi que les mots clés trouvés manuellement grâce à la recherche, et suivant les indications SEO (Optimisation pour les moteurs de recherche). Nous allons ensuite découper le texte en grandes idées en vue de la rédaction d'une pige journalistique, dont chacune devient alors un ensemble que nous allons traiter ensuite. Nous déterminons ensuite les balises

FIGURE 2.1 – Principales étapes d'un projet avec Scribt



qui vont être intégrées au projet pour appeler les données. Egalement, les données d'entrées sont prétraitées, de manière à pouvoir être utilisées convenablement par la suite.

Production avec le backoffice

Pour chaque grande idée du texte, il est nécessaire de rédiger un certain nombre de phrases, qui pourront alors entraîner le logiciel dans l'univers sémantique correspondant. Aussi, pour chacune des données d'entrée, il est nécessaire de créer une balise. Cette balise contiendra la donnée associée, ainsi que les règles qui la composent. Nous pouvons alors déterminer sa nature, son genre et les éventuels traitements. En effet, il est nécessaire de déterminer si la donnée peut être présente dans tous les textes, ou si elle ne doit sortir que dans un cas particulier : si une autre donnée est présente, par exemple.

Les modèles sont ensuite entrés dans le backoffice, où nous pouvons ajouter des balises de synonymes. Ces balises sont ajoutées aux phrases, afin d'y intégrer de la synonymie. Ainsi, nous pouvons ajouter de la variabilisation au corpus de modèles.

Production avec l'interface d'administration

Une fois les phrases intégrées dans le backoffice, nous pouvons passer sur l'outil d'administration des projets. Nous pouvons configurer de nombreux paramètres de notre système dans cet outil. En effet, l'intégration des balises contenant les données d'entrée se fait ici, cependant à ce stade elles sont encore vides. Nous indiquons donc les règles propres à chacune des balises, et nous les remplissons en indiquant quelle

donnée d'entrée elles doivent contenir.

Il y a plusieurs possibilités de règles concernant les balises, nous pouvons jouer avec le format des données d'entrée, de manière à obtenir un corpus qui convient parfaitement à nos modèles.

Nous pouvons également gérer le plan de nos textes finaux. En effet, il est possible d'avoir plusieurs genres de textes à générer en fonction du projet. Pour les petites annonces immobilières notamment, en fonction des biens concernés, il est possible de faire un plan par type de bien. Nous pouvons également introduire des règles pour notre/nos plan(s). Par exemple, il est possible de déterminer la sortie des phrases en fonction des données d'entrées obtenues. Cette fonctionnalité est utile notamment dans le cas où, en fonction de la valeur contenue dans une balise, nous désirons utiliser une phrase si la valeur de la balise est A, et une autre phrase si la valeur est B.

Débogage / Tests

Une fois les réglages effectués, il est possible de générer nos premiers textes. Les premiers essais, notamment quand les données d'entrées sont importantes et les réglages nombreux, sont parfois insatisfaisants. Il est nécessaire d'effectuer un débogage de toutes les erreurs rencontrées. Pour cela, nous pouvons utiliser le logiciel Antidote, pour repérer les erreurs d'orthographe ou de syntaxe. Une fois les premières corrections effectuées, la phase de test/évaluation peut être lancée. Nous partageons nos textes générés à des relecteurs humains, qui pourront voir des erreurs qui nous auraient échappé, ou des tournures de phrases qui pourraient être revues. Un document est ainsi créé pour que les relecteurs puissent indiquer les erreurs vues, qui seront alors corrigées. Nous avons par ailleurs rédigé une catégorisation des erreurs rencontrées au cours des projets effectués (voir le chapitre Résultats).

Le temps nécessaire à l'exécution de ce type de projet oscille entre 1 mois pour les projets les plus courts, à plusieurs mois pour les projets complexes.

2.3 Discussion

2.3.1 Définition d'un projet

Nous avons vu dans l'état de l'art diverses méthodes de GAT, avec diverses architectures. En effet, en fonction du format des données d'entrée ainsi que du but recherché par la génération de texte, il est important de choisir la méthode adéquate. Cependant, dans le cas du système de GAT de Labsense, il est possible de générer des textes pour tout type de contexte : descriptions de produits, rédaction de textes météorologiques, annonces immobilières, résultats de matchs sportifs, etc. Cette polyvalence est due au type de système de GAT que nous avons ici, à savoir un système avec modèles (templates). En effet, ce genre de système dépend en grande partie de la participation humaine, avec la rédaction de modèles adaptés et la préparation minutieuse des données d'entrée.

Ainsi, il sera impératif de se renseigner sur le genre et le ton à adopter pour les textes générés, car cela sera différent pour chaque type de projet. Les règles ainsi que les modèles de phrases ne sont pas réutilisables, à moins d'avoir des projets très similaires, ou des données d'entrée identiques. Les expressions régulières (regular expression) souvent employées dans la définition des règles peuvent être certes utilisées dans plusieurs cas. Cependant, pour chaque projet de génération de textes, la production doit être remise à zéro, occasionnant donc une perte de temps assez considérable. Il existe certes des modèles de phrases qui peuvent être utilisés dans plusieurs contextes, mais cela reste tout de même rare.

Nous avons donc décidé de prendre comme base de notre mémoire de recherche un projet de génération automatisée de textes contenant des données immobilières : en effet, nous avons produit ce projet dans sa globalité dans le cadre du stage. Ces textes ont pour objectif de donner des informations de type immobilières pour des villes, dans le but d'aider peut-être à l'achat d'une maison ou d'un appartement dans le département concerné. Ces textes sont donc à destination des personnes désireuses d'avoir des informations, et qui sont prêtes à effectuer un achat immobilier.

2.3.2 Définition d'une problématique de recherche

Nous avons, dans le cadre de notre stage, fait l'expérience d'un projet de GAT avec modèles, avec une évaluation à visée commerciale très basique. Après les recherches effectuées dans le cadre de l'état de l'art, nous avons envisagé la possibilité de tester d'autres systèmes de GAT, dans le but de produire une comparaison avec le système de GAT avec modèles. En effet, nous pouvons nous demander s'il est possible de trouver un système nécessitant une préparation humaine moindre, avec des résultats équivalents. Nous avons ainsi choisi de tester les systèmes de GAT dotés de réseaux neuronaux, car nous avons vu dans l'état de l'art qu'ils ne nécessitaient que peu de travail humain pour fonctionner. Cela nous semble être une opposition intéressante : un système de GAT comptant beaucoup sur le travail humain, tandis que pour l'autre système de GAT, beaucoup moins. Également, nous pensons également produire une évaluation complète, selon des critères choisis grâce à nos recherches.

Nous allons donc chercher quelles sont les spécificités des systèmes de GAT utilisés dans ce mémoire, les préparatifs nécessaires des données d'entrée et une évaluation adéquate pour ces mêmes systèmes.

2.4 Conclusion

Dans ce mémoire, nous allons tout d'abord décrire le processus de création d'un projet avec le système de GAT avec modèles utilisé dans le cadre de notre stage, puis avec un système de GAT avec réseau neuronal. Nous allons ensuite présenter les résultats obtenus, et effectuer un comparatif de ces deux systèmes. Enfin, nous allons produire un protocole d'évaluation humaine dans le but de déterminer si les textes obtenus remplissent les critères que nous avons choisis.

Deuxième partie

Expérimentations

MÉTHODES

Sommaire

3.1	Introduction	31
3.2	Mise en place d'un système de GAT avec modèles	31
3.2.1	Création des modèles	31
3.2.2	Intégration des modèles au programme	35
3.2.3	Préparation du corpus de données d'entrée	36
3.2.4	Réglages des différents corpus	38
3.2.5	Conclusion	41
3.3	Mise en place d'un système de GAT avec Deep Learning	41
3.3.1	Outils	41
3.3.2	Corpus	41
3.3.3	Configuration du modèle	42
3.3.4	Conclusion	42
3.4	Conclusion	43

3.1 Introduction

Au cours de notre stage, nous avons eu la charge d'un projet de génération de textes sur le sujet de l'évolution des prix de l'immobilier. Ce projet est un projet créé dans l'optique de servir de démonstration. Nous avons donc cherché les données d'entrée nous-mêmes, de manière la plus détachée possible, pour avoir un corpus neutre sur lequel tester le modèle créé. Idéalement, les données pourraient provenir de sources comme des données en accès libre et gratuit (données INSEE par exemple). Nous vous présenterons donc la mise en place de ce projet, ainsi que l'expérience qui a été accomplie en comparaison à celui-ci.

3.2 Mise en place d'un système de GAT avec modèles

3.2.1 Création des modèles

Appréhension du sujet et rédaction de texte cible

Des recherches précises ont été effectuées en amont de la rédaction du texte cible visible ci-dessous. En effet, il est important de se documenter un maximum avant de débiter un projet, pour en comprendre pleinement la problématique et pouvoir proposer des textes adéquats. La recherche sémantique est donc primordiale, et s'accompagne d'outils tels que Semrush, ou simplement Google avec les mots clés obtenus

par le logiciel précédemment nommé. Pour ce projet, nous voulons des données sur le prix des maisons et des appartements, dans l'immobilier neuf comme dans l'ancien. Ces données doivent concerner une ville d'un département, et comparer les prix de cette année avec ceux des années précédentes. Le tout doit être dans un registre accessible au plus grand nombre, mais doit tout de même comprendre des termes précis du domaine de l'immobilier.

Dans le domaine de la GAT, les données d'entrées sont importantes mais le registre souhaité pour la génération des textes ainsi que le domaine de spécialité concerné sont des éléments à prendre en compte dès l'appréhension du projet.

A Versailles, le volume de transaction dans l'ancien est en hausse de 11 % depuis janvier 2020, avec des prix globalement orientés à la hausse : +3,7 % pour les maisons, et +2,65 % pour les appartements par rapport à l'année précédente. En 2019, le prix du m² à Versailles s'élevait à 3 200 € en moyenne, pour 3 500 € aujourd'hui. Les appartements type T3 se sont vendus 4 % plus chers en 2019 qu'en 2018. C'est le type de bien qui a pris le plus de valeur ces dernières années. A l'inverse, la valeur des T6 a chuté (2,5 % moins chers entre ces deux mêmes années).

En revanche, le neuf peine davantage et subit une baisse des prix depuis quelques années. Pour une maison, le prix du m² neuf était de 4 150 € en 2016, alors qu'il est aujourd'hui de 3 900 €. Il en va de même pour les appartements, qui subissent une chute de 5,7 % depuis 2019 (3 800 € / m² en 2019 pour 3 554 € aujourd'hui).

Versailles suit les tendances des Yvelines. En moyenne, en 2019, les appartements et maisons yvelinoises se vendaient à 3 856 € du m² neuf, ou encore 3 741 € dans l'ancien. Tandis que le prix des appartements a légèrement augmenté (+1,7 %), les maisons ont subi une évolution différente et leur prix a chuté de 2,2 % en un an. La ville du département qui connaît la plus grosse hausse de prix est Feucherolles : +4,2 % entre 2019 et 2020. A l'inverse, c'est à Trappes que les prix ont le plus chuté entre ces périodes (-5,4 %).

TABLE 3.1 – Texte cible du projet "Evolution des prix de l'immobilier"

Nous partons donc de ce texte cible ci-dessus, rédigé pour remplir les potentiels besoins du client. Les données intéressantes à retrouver dans les textes générés doivent être insérées dans ce texte cible. En effet, c'est à partir de ce texte et des données qu'il contient que le projet sera exécuté. Lors d'un projet pour un client, le volume du texte ainsi que les données contenues sont décidés en accord avec ses demandes. Pour ce projet de démonstration, ces paramètres ont été choisis en accord avec le résultat des recherches effectuées sur le sujet, ainsi que du volume de données qui semblent importantes. Il vaut mieux avoir trop de données en entrée que pas assez : en effet, il est toujours possible de régler les textes générés pour ne pas sortir une donnée, alors que pour ajouter une donnée il faut créer un nouveau modèle de phrase, et refaire les réglages.

Quand le texte cible a été validé, nous commençons par mettre en relief les données qui seront considérées comme les données d'entrée : c'est-à-dire dans le cadre de ce projet, les noms de ville, les types d'appartements, les pourcentages et les années.

A **Versailles**, le volume de transaction dans l'ancien est en hausse de **11 %** depuis **janvier 2020**, avec des prix globalement orientés à la hausse : **+3,7 %** pour les maisons, et **+2,65 %** pour les appartements par rapport à l'année précédente. En **2019**, le prix du m² à **Versailles** s'élevait à **3 200 €** en moyenne, pour **3 500 €** aujourd'hui. Les appartements type **T3** se sont vendus **4 %** plus chers en **2019** qu'en **2018**. C'est le type de bien qui a pris le plus de valeur ces dernières années. A l'inverse, la valeur des **T6** a chuté (**2,5 %** moins chers entre ces deux mêmes années).

En revanche, le neuf peine davantage et subit une baisse des prix depuis quelques années. Pour une maison, le prix du m² neuf était de **4 150 €** en **2016**, alors qu'il est aujourd'hui de **3 900 €**. Il en va de même pour les appartements, qui subissent une chute de **5,7 %** depuis **2019** (**3 800 € / m²** en **2019** pour **3 554 €** aujourd'hui).

Versailles suit les tendances des **Yvelines**. En moyenne, en **2019**, les appartements et maisons **yvelinoises** se vendaient à **3 856 €** du m² neuf, ou encore **3 741 €** dans l'ancien. Tandis que le prix des appartements a légèrement augmenté (**+1,7 %**), les maisons ont subi une évolution différente et leur prix a chuté de **2,2 %** en un an. La ville du département qui connaît la plus grosse hausse de prix est **Feucherolles** : **+4,2 %** entre **2019** et **2020**. A l'inverse, c'est à **Trappes** que les prix ont le plus chuté entre ces périodes (**-5,4 %**).

TABLE 3.2 – Texte cible du projet "Evolution des prix de l'immobilier" avec les données mises en valeur

Pour chaque donnée mise en valeur, nous allons créer une balise dans un format adapté. Egalement, il y a certaines règles à respecter : en effet, si la donnée contenue dans la balise voit son genre/nombre modifié via un traitement dans la partie backoffice, la modification apparaîtra dans le nom de la balise. Il est donc nécessaire de ne pas entrer de nom de balise qui laisse à penser que la modification a déjà été effectuée.

Nous allons donc créer un tableau contenant les noms de balise, et les règles qui les définissent. Cette étape est primordiale, ce sont les fondations de notre projet.

Ainsi, nous avons toutes les informations nécessaires à portée de main et nous allons pouvoir les utiliser lors de la fusion du corpus de modèles et du corpus de données d'entrée.

Découpage du texte et mise en place des parties

Pour cette partie du projet, nous prenons le texte cible et le partageons en plusieurs parties, selon les idées qui en ressortent. Ces parties seront nos modèles pour la rédaction. En effet, il sera nécessaire de faire figurer les mêmes informations dans chacune des phrases que nous allons rédiger. Il peut y avoir deux phrases pour une même grande idée, il sera juste nécessaire d'être vigilant lors de la rédaction de pape pour la partie concernée.

Rédaction des modèles

Lors de la rédaction, nous avons dû faire attention à plusieurs éléments qu'il a fallu reproduire dans toutes les phrases écrites :

- le registre ;
- le format des données d'entrée ;
- l'intégralité des données d'entrée ;

En effet, pour une grande idée, les phrases doivent être différentes mais être tout de même similaires pour ces 3 éléments.

En moyenne, en 2019, les appartements et maisons yvelinoises se vendaient à 3 856 € du m ² neuf, ou encore 3 741 € dans l'ancien.
En 2019, on peut noter que les appartements et maisons yvelinoises se vendaient à 3 856 € le m ² neuf, et 3 741 € pour l'ancien en moyenne.
Les appartements et maisons yvelinoises se vendaient en moyenne à 3 856 € le m ² en 2019 en ce qui concerne le neuf et 3 741 € pour l'ancien.

TABLE 3.3 – Exemple de modèles pour une partie du texte cible

Comme nous pouvons le voir sur le tableau ci-dessus, contenant les phrases rédigées pour la partie concernée, les phrases ont une forme sensiblement différente, mais apportent les mêmes informations, elles ont la même factualité.

Il est primordial de garder le même format pour les données d'entrée. Sinon, la règle associée à cette donnée sera obsolète et le texte généré en sortie sera déficient.

Il y a certains cas particuliers parmi les attributs visibles sur le tableau. En effet, quelques phrases présentent une certaine inclinaison positive ou négative par rapport aux données d'entrée. Pour la partie sur la tendance des prix dans l'immobilier neuf notamment :

« En revanche, le neuf peine davantage et subit une baisse des prix depuis quelques années. »

Cette phrase fonctionne dans le cas où les prix de l'immobilier neuf sont en baisse, mais pas dans le cas contraire. Il a donc fallu créer des variations de modèles.

Tendance Neuf	
Hausse	Le neuf quant à lui est en pleine expansion, avec des prix qui augmentent ces dernières années.
Baisse	Avec une baisse des prix ces dernières années, le neuf est en perte de vitesse.

TABLE 3.4 – Exemple de création de sous-parties

Insertion des balises dans les modèles

Quand la phase de rédaction est terminée, il est nécessaire d'insérer les balises dans les phrases.

En moyenne, en [ANNEE], les appartements et maisons [GENTILE] se vendaient à [PRIX 1] du m ² neuf, ou encore [PRIX 2] dans l'ancien.

TABLE 3.5 – Exemple d'insertion de balises dans les modèles réalisés pour une des parties

Il est possible que la présence d'une des données d'entrée ne soit pas nécessaire dans la phrase. Dans ce cas, il sera nécessaire d'effectuer une opération pour rendre facultative la balise.

Jusqu'ici, nous avons mis en place les modèles, c'est-à-dire les phrases qui contiendront les données d'entrée. Nous avons donc rédigé plusieurs phrases, qui seront insérées dans le backoffice. Ces phrases constituent la base de notre projet. En effet, comme nous l'avons vu dans l'état de l'art, les systèmes de GAT à base de modèles (templates) sont constitués de deux éléments : les phrases « réceptacles » et les données qui seront insérées dans ces phrases. Nous avons donc un de ces deux éléments, auquel nous allons ajouter de la variabilité. Effectivement, nous n'avons pas un assez grand choix de phrases pour générer des textes sensiblement différents à ce stade du projet. C'est pourquoi l'étape de variabilisation mise en place dans la partie Backoffice est primordiale.

3.2.2 Intégration des modèles au programme

Réflexion pré-intégration

Lors de cette étape du projet, il ne s'agit pas d'uniquement entrer les phrases dans le logiciel, il faut également insérer de la variabilité dans ces phrases. En effet, avec le backoffice nous pouvons remplacer un mot par une balise qui en contient plusieurs. Cela permet ainsi de maximiser l'aspect aléatoire des textes générés. Il faut cependant faire attention à plusieurs choses :

- conserver le style du texte d'origine ;
- rester dans le même champ lexical : ici l'immobilier ;
- faire les bons choix de synonymes pour la variabilisation ;

Tout en gardant en tête ces éléments, nous pouvons donc intégrer nos phrases.

Intégration dans le backoffice

Phrase de base	La ville du département qui connaît la plus grosse hausse de prix est Feucherolles : + 4,2 % entre 2019 et 2020.
Phrase avec balises de données d'entrée	La ville du département qui connaît la plus grosse hausse de prix est [VILLE] : [TAUX] entre [ANNEE MIN] et [ANNEE MAX].

TABLE 3.6 – Exemple de phrase avec les différentes balises

Dans ce tableau nous pouvons voir les phrases dans des stades que nous avons abordés précédemment. A la suite de cela, le backoffice est capable de gérer une variabilisation synonymique au coeur même des phrases.

En effet, l'insertion des synonymes dans les modèles permet d'atteindre un niveau de variabilisation rendant les textes finaux uniques. Un groupe nominal originel « la plus grosse hausse de prix » pourra donc se changer en plusieurs autres groupes nominaux tels que :

- « la plus forte augmentation de prix » ;
- « la plus importante hausse de prix » ;
- « la plus grosse augmentation de prix » ;

Le sens de cette partie de phrase est bien conservé, mais nous avons à présent plusieurs façons de transmettre l'information. Pour une seule phrase parmi notre corpus, nous avons déjà un grand nombre de possibilités de génération.

Le backoffice gère également les accords entre les différents mots de la phrase.

A l'issue de cette partie, nous avons pu créer notre corpus de phrases modèles (templates), et insérer de la variabilisation dans ce corpus. En effet, les textes générés seront ainsi différents les uns des autres, grâce à toutes les possibilités de génération dont nous disposons à présent. Nous devons à présent constituer le corpus de données d'entrée, c'est-à-dire la deuxième partie de notre système de GAT.

3.2.3 Préparation du corpus de données d'entrée

Réflexion pré-préparation

Comme dit précédemment, ce projet a une visée de démonstration, et donc pas de corpus de données d'entrées fourni par un client. Il est donc nécessaire de créer ce corpus de la manière la plus neutre possible. Pour nous aider à faire cela, nous devons tout d'abord créer un tableau destiné au potentiel client, contenant la liste de toutes les catégories de données nécessaires, leur type (string, int ou list), une description des données, leur valeur possible (si ce sont des listes), s'il est possible de sortir plusieurs données pour une catégorie et si cette catégorie est obligatoire. Il s'agit d'être le plus clair possible sur cette étape, car il est nécessaire de pouvoir

savoir exactement quel type de données sera transmis pour l'étape d'insertion dans les modèles.

Données d'entrée

Catégorie	Type	Description	Valeur possible (si list)	Plusieurs choix possibles (si list)	Obligatoire
Ville	string	Nom de la ville concernée			oui
Type d'appartement - meilleur taux	list	Type d'appartement qui possède le meilleur taux de cette année	Studio, T1, T2, T3, T4, Loft, Duplex, Souplex	non	oui
Prix du m ² actuel	int	Prix moyen du m ² pour l'année actuelle			oui
Année de comparaison – taux appartement neuf	int	Année de comparaison (avec année actuelle) pour l'évolution du prix d'un appartement neuf			oui
Taux actuel – appartements anciens	int	Taux pour les appartements anciens pour l'année actuelle			oui

TABLE 3.7 – Exemple de tableau regroupant les informations sur les données d'entrée requises

Nous pouvons ainsi voir trois types de données, les chaînes de caractères (string), les nombres (int) et les listes à choix (list). Après avoir rédigé ce tableau, nous pouvons créer notre tableau de données d'entrées, en prenant le tableau ci-dessus comme référence.

Ville	Type d'appartement - meilleur taux	Description	Année de comparaison - taux appartement neuf	Taux actuel - appartements anciens
Versailles	T1	2 900 €	2015	2 %
Lille	T2	3 200 €	2017	- 6 %
Toulouse	T3	3 600 €	2016	- 2,8 %
Le Mans	Studio	2 600 €	2016	- 3 %

TABLE 3.8 – Exemple de données pour les catégories présentées dans le tableau précédent

Les données de la colonne « Type d'appartement – meilleur taux » ont été choisies à partir d'une liste déjà créée, cela permet de contrôler la forme des données d'entrées. Pour les données contenant des nombres comme les prix ou les taux, nous pouvons choisir un format incluant les unités comme l'euro, ou ne prendre que les nombres. Si nous ne prenons que les nombres, il faudra penser à créer une règle de concaténation pour ajouter l'unité dans l'interface d'administration de projet. De même, les données concernant les taux devront être réglées de manière adéquate.

Données annexes

Dans certains cas, nous avons besoin de créer d'autres données à insérer dans les phrases. Nous en avons besoin notamment pour les phrases de ce genre :

En moyenne, en 2019, les appartements et maisons yvelinoises se vendaient à 3 400 € du m² neuf, ou encore 3 200 € dans l'ancien.

En effet, nous obtenons l'information du département concerné via les données d'entrée. De cette information va dépendre l'adjectif utilisé dans ce modèle de phrase, ici « yvelinoises ». Nous allons donc créer d'autres données, qui contiendront d'un côté les noms de département, et de l'autre l'adjectif à la forme féminin pluriel. De cette manière, si nous avons tel département, nous pourrons donc l'associer à tel adjectif.

Nous devons ensuite faire les réglages nécessaires sur l'interface d'administration pour utiliser de manière efficace ces données.

3.2.4 Réglages des différents corpus

Avant de pouvoir entrer directement les données d'entrée dans l'interface d'administration des projets, il faut créer les réceptacles de ces données. Nous intégrons donc la liste des balises utilisées dans le projet dans l'interface. Comme expliqué précédemment, nous devons donc également entrer toutes les informations concernant ces balises.

Une fois les balises renseignées sur cette interface, nous pouvons entrer le fichier contenant les données d'entrée sur l'interface d'administration.

Ces données sont brutes, c'est-à-dire qu'elles n'ont pas encore été modifiées par les règles que nous avons rédigées lors de la pré-production du projet.

Concernant les autres données, nous avons vu plus haut la création d'un document annexe pour les départements français, mais nous avons également créé d'autres documents. En effet, nous avons rencontré quelques problèmes pendant le déroulement de ce projet qu'il a fallu résoudre de cette manière.

Concernant notre corpus, nous avons plusieurs types de données d'entrée :

- des années ;
- des noms de ville ou de département ;
- des types d'appartements ;
- des taux ;

Les taux forment à eux seuls 50 % du corpus de données d'entrée, ils ont donc une certaine importance. Nous avons plusieurs genres de phrases modèles, certaines contenant des taux avec un signe plus ou moins, d'autres avec des taux sans signe.

Taux de la ville du département qui ont le plus augmenté	Taux de la ville du département qui ont le plus descendu
5 %	-3,9 %
9 %	-5 %
5 %	-10 %
15 %	-2,2 %
13 %	-8,5 %
3,7 %	-6,4 %

TABLE 3.9 – Exemple de donnée de taux avec ou sans signes

La différence entre ces deux types de taux est primordiale dans les modèles de phrases. En effet, la présence de signe pour certains taux appuie le fait qu'il s'agit d'une baisse ou d'une hausse, alors que dans d'autres phrases le signe paraîtrait déplacé du fait de la présence de l'adjectif « plus cher » juste après la donnée du taux, la même information serait ainsi signifiée deux fois, ce qui serait redondant. Nous avons donc dû opérer une distinction de ces deux types de taux, ce qui a donné deux types de balises contenant des taux.

Les données clients récupérées contenaient des taux qui comportaient tous des signes, nous avons donc dû entrer ces données dans des balises, et en créer d'autres qui contenaient également une règle qui indique que le signe doit être retiré.

Nous avons utilisé des expressions régulières (regex ou regular expression) pour retirer les signes + et -. Ainsi, nous disposons des deux types de balises, que nous pouvons utiliser librement dans les modèles en fonction du contexte.

Nous devons également effectuer un réglage pour pouvoir trouver le bon nom de gentilé, celui qui sera associé au nom de département présent dans nos données d'entrées. En effet, en fonction du département en question, nous devons insérer le bon adjectif dans le modèle. Nous pouvons donc effectuer cette manipulation sur l'interface d'administration, pour trouver l'adjectif de manière automatique.

Nous obtiendrons ainsi un résultat de ce genre :

Ville	Département	Adjectif de gentilé au féminin pluriel
Le Mans	Sarthe	sarthoises

TABLE 3.10 – Exemple de résultats souhaités pour le cas du gentilé

Comme nous pouvons le voir ici, pour un des textes générés nous aurons bien les 3 informations concernant la ville concernée et son département, dans le format attendu.

Pour les parties ayant des sous-parties comme nous avons vu précédemment, il y a également des réglages spécifiques et des groupes de mots à remplir avec ce genre de manipulations. La partie la plus complexe à régler est celle de la phrase d'introduction du texte. En effet, les modèles de phrases qui composent cette grande idée se scindent en deux parties, la première mentionne la hausse ou la baisse des prix dans l'immobilier ancien, tandis que la deuxième fait mention de la hausse, la baisse, ou le maintien des prix.

Sous-partie	Phrase modèle exemple
Hausse des prix dans l'ancien / hausse des prix des maisons et appartements	A [VILLE], le volume de transaction dans l'ancien est en hausse de [TAUX 1] depuis [ANNEE], [TENDANCE DES PRIX] : [TAUX 2] pour les maisons, et [TAUX 3] pour les appartements par rapport à l'année précédente.
Baisse des prix dans l'ancien / baisse des prix des maisons et appartements	On peut constater à [VILLE] une baisse du volume de transaction de [TAUX 1] dans l'ancien, [TENDANCE DES PRIX]. En effet, pour les maisons on a un taux de [TAUX 2], et de [TAUX 3] pour les appartements depuis [ANNEE].

TABLE 3.11 – Exemple de sous-parties créées pour la phrase introductive

Il y a donc plusieurs paramétrages à effectuer pour ce modèle : il faut remplir de manière convenable la partie qui indique la tendance des prix, et assurer l'utilisation de la sous-partie adéquate en fonction des taux positifs ou négatifs utilisés. Nous avons donc réalisé plusieurs règles pour répondre à ce problème.

L'utilisation d'expressions régulières a encore une fois été nécessaire, pour parvenir à repérer les taux qui étaient précédés d'un signe positif ou négatif. Cela permet ainsi d'attribuer un résultat adéquat et donc de sortir la bonne sous-partie dans le texte généré. En effet, si le taux de prix de l'immobilier ancien est précédé d'un signe plus, nous allons pouvoir indiquer qu'il faut générer la phrase parlant de

taux positifs.

Nous pouvons également nous servir de cette logique pour indiquer quelle sous-partie il faudra générer, parmi celles que nous avons vues précédemment. Nous utilisons donc une règle qui va choisir la sous-partie correspondante. En effet, si la donnée contenue dans la balise n'est pas égale à la chaîne de caractères indiquée, qui est une expression régulière (qui indique donc une hausse ou une baisse), la sous-partie ne sera pas utilisée. De cette manière, nous assurons l'utilisation de la bonne sous-partie.

Nous avons d'autres cas de sous-parties à gérer pour ce projet, mais leurs réglages étant identiques en tous points à ceux que nous venons d'aborder, ils ne seront pas indiqués ici.

3.2.5 Conclusion

Nous avons donc constitué nos deux corpus nécessaires au fonctionnement de ce système de GAT avec modèles : le corpus de phrases modèles et le corpus de données d'entrées. Nous avons également lié ces corpus grâce à des règles et des paramètres, dans le but d'assurer une génération fonctionnelle et sans problèmes.

3.3 Mise en place d'un système de GAT avec Deep Learning

3.3.1 Outils

Pour comparer les résultats de la GAT avec des modèles, nous avons effectué des tests de génération avec cette fois-ci un système de génération automatique de textes avec une architecture en réseau de neurones (Deep Learning). Pour se faire, nous utilisons donc **textgenrnn**, un module python créé par [Woolf, 2017] utilisant Keras et Tensorflow. Cette architecture en réseau de neurones utilise des techniques telles que la pondération de l'attention et le skip-embedding (la prédiction de mot via plongement lexical) pour permettre un entraînement rapide des données. De plus, ce module nous permet de gérer nous-mêmes la taille du réseau neuronal, le nombre des couches et le nombre de cycles, d'itérations d'apprentissage (que nous appellerons epoch dans ce document).

3.3.2 Corpus

Concernant le corpus de données d'entrée pour cette expérimentation, nous avons d'abord pensé utiliser le texte cible qui avait été rédigé pour le projet de GAT avec modèles. Cependant, nous redoutons que cela soit trop peu. En effet, pour entraîner correctement un système de réseau de neurones, il doit y avoir un nombre conséquent de données. Il serait alors préférable de donner à ce programme des textes multiples, pour espérer avoir des résultats corrects. Néanmoins, nous avons décidé de tester notre système avec plusieurs corpus de différentes tailles, pour pouvoir observer la différence que la taille du corpus peut créer sur les textes générés. Nous avons donc créé 3 corpus :

- un corpus contenant uniquement le texte cible ;
- un corpus contenant 10 textes issus du système de GAT avec modèles ;
- un corpus contenant 30 textes issus du système de GAT avec modèles

Nous avons fait en sorte de choisir des textes avec des données d'entrée différentes, pour avoir une plus grande variabilité.

3.3.3 Configuration du modèle

Avant de pouvoir générer des textes grâce au modèle, il faut d'abord l'entraîner. C'est une étape cruciale pour les programmes utilisant les réseaux de neurones. En effet, il s'agit de donner un certain pourcentage des données d'entrée au système, pour qu'il entraîne un modèle. Ce même modèle sera lui testé sur le pourcentage de données d'entrée restantes, en prenant en compte les paramètres qui ont été décidés pendant l'entraînement. Nous avons donc la possibilité de décider du nombre d'itérations, de cycles d'apprentissage qu'effectuent les données d'entrée dans le réseau de neurones pendant l'entraînement du modèle, soit des epochs.

Pour les tests, nous avons décidé de lancer l'entraînement avec 50 epochs. Cependant, nous nous sommes vite aperçus que 50 epochs n'étaient pas suffisantes pour laisser le temps au programme d'appréhender le corpus. Nous avons décidé de faire des tests supplémentaires avec 100 et 200 epochs.

Nous avons également décidé d'effectuer 2 types d'expérimentations, une avec les paramètres par défaut du réseau de neurones et une avec des paramètres que nous aurons définis. Pour le modèle personnalisé, nous avons décidé de lancer un entraînement au niveau des caractères et non des mots, d'avoir un réseau composé de 4 couches qui ont chacune 128 neurones, et de limiter le nombre de tokens à considérer avant de prédire un mot à 8 tokens. Au niveau de l'entraînement, nous avons précisé que nous voulions la génération d'un texte exemple entre chaque epoch, pour visualiser directement l'évolution de l'apprentissage, et nous avons décidé de prendre 90 % des données d'entrées pour faire cet entraînement.

3 scripts ont donc été écrits, un contenant la configuration expliquée dans le paragraphe ci-dessus, un autre pour entraîner le modèle et le dernier script pour lancer une génération de textes à différentes températures qui prend en compte le modèle créé lors de l'étape de l'entraînement. La température, comprise entre 0,2 et 1, nous permet de « jouer » avec la créativité des résultats que nous obtenons, les résultats les moins créatifs étant ceux de la température 0,2.

3.3.4 Conclusion

Après avoir pu expérimenter le système de GAT avec modèle au cours de notre stage, nous avons pu également produire un système de GAT avec réseau neuronal. Nous avons ainsi constitué les corpus nécessaires, mis en place la configuration du réseau de neurones et effectué des tests pour déterminer quelle est la meilleure configuration pour avoir des résultats corrects.

3.4 Conclusion

Le temps de réalisation de ces deux projets sur les deux systèmes de GAT présentés plus haut est relativement différent. Effectivement, le système de GAT avec modèles est un système qui repose encore beaucoup sur le travail fourni par les mains humaines. En effet, l'écriture du texte cible, puis des modèles, ainsi que la mise en place des règles et paramétrages multiples sont de longs processus. La charge de travail humaine est assez considérable. Cependant, le temps passé sur le projet et le travail fourni par des humains a pu garantir une excellente qualité des textes générés. Les différentes étapes du projet étant soigneusement respectées, avec plusieurs phases de test, le risque d'obtenir des textes incorrects grammaticalement parlant est en effet très faible. La rédaction des modèles, la mise en forme des données, la mise en place des règles et paramétrages, la validation de personnes de l'entreprise, tout cela a pris du temps. Concernant le système de GAT avec réseau de neurones, la partie de réglages du réseau a été relativement courte. En effet, le module `textgenrrn` permet de modifier les paramétrages de manière aisée, comme démontré dans l'annexe du code du script `config.py`. Nous pouvons ainsi régler le nombre de neurones, ainsi que le nombre de couches et d'autres paramètres rapidement. Cependant, en fonction du nombre d'époches indiqué, l'entraînement et la génération des textes sont très longs. Pour l'entraînement de 100 epochs sur les données, cela nous a pris environ 5 heures pour obtenir un modèle, et encore 3 heures pour obtenir les textes générés. Il y a donc un temps d'attente assez long avant de pouvoir observer les premiers résultats du système.

RÉSULTATS

Sommaire

4.1	Introduction	45
4.2	Système de GAT avec modèles	45
4.2.1	Phase de débogage	45
4.2.2	Catégorisation des erreurs	46
4.2.3	Textes obtenus	47
4.3	Système de GAT avec réseau neuronal	47
4.3.1	Essais	47
4.4	Conclusion	51

4.1 Introduction

Dans ce chapitre, nous présentons les textes obtenus avec les deux systèmes de GAT précédemment expliqués. Pour le système de GAT avec modèles, nous ne présenterons que les textes après correction des erreurs. Cependant, pour le système de GAT avec réseau neuronal, nous avons dû faire de nombreux essais pour trouver la configuration idéale, nous montrerons donc nos essais infructueux également, dans le but d'observer une amélioration des résultats.

4.2 Système de GAT avec modèles

4.2.1 Phase de débogage

Quand tous les réglages ont été effectués sur l'interface d'administration pour les balises et les modèles de phrases, nous avons procédé à une vérification des premiers textes générés. Il s'agit de vérifier si les réglages et les données d'entrées apparaissent bien dans les textes. Nous nous assurons ainsi de la qualité de la fusion du corpus de phrases modèles avec les données d'entrée. Une fois les vérifications de base terminées, nous produisons un document de test : il contient plusieurs colonnes qui indiquent des informations nécessaires à la correction des erreurs, comme un descriptif de l'erreur en question et son type (bug ou amélioration possible) ainsi que des informations permettant de retrouver l'erreur parmi les textes. La relecture de ces textes par plusieurs personnes permet ainsi de détecter les fautes que nous n'aurions pas aperçues car ayant travaillé des semaines sur le projet et n'ayant donc pas assez de recul.

Type d'erreur	Description de l'erreur	Exemple	Statut
bug	Le pronom associé par défaut au nom (balise) n'a pas été retiré	Dans la ville de Marseille, on peut noter que le volume de transaction dans l'immobilier ancien est en l'une hausse de 5 %	corrigé
bug	La contraction de la préposition avec le pronom de la balise ne se fait pas	A Le Mans , une diminution de 10 % du volume de transaction dans l'ancien est perceptible	corrigé

TABLE 4.1 – Exemple de document de test pour le projet "Evolution des prix de l'immobilier"

4.2.2 Catégorisation des erreurs

Tout au long de notre période de stage, nous avons eu l'opportunité de tester et de déboguer plusieurs projets. Nous avons constaté que les erreurs relevées dans les textes générés avec ce système de GAT étaient souvent similaires. Nous avons donc rédigé une catégorisation des erreurs les plus courantes.

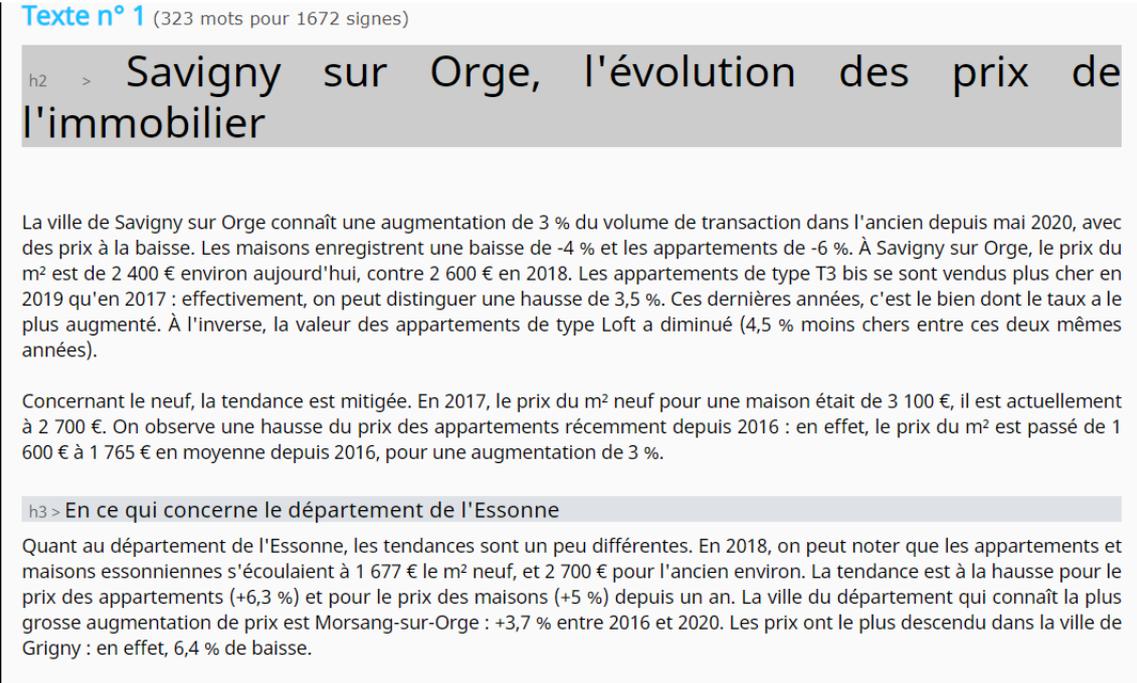
Type d'erreur général	Erreurs
Erreurs d'accord	l'accord n'a pas été indiqué
	l'accord a été mal fait
	la balise a été mal accordée
Balises absentes du texte final	la balise a été mal remplie
	La balise a été mal paramétrée
Phrases absentes du texte final	l'attribut a été mal paramétré
	La balise est absente (dans le cas où l'attribut est utilisé seulement quand la balise est présente)
Répétition de mot	un adjectif pronominal qui est présent dans la phrase et dans la balise
	une balise ([NOM], [ADJECTIF] par exemple) qui fait doublon avec un terme fixe du modèle
Contenu des balises qui ne correspond pas	Factorisation du contenu d'une balise non réussie
	la balise a été mal remplie

TABLE 4.2 – Catégorisation des erreurs dans les textes générés via modèle

Comme nous pouvons le voir dans le tableau de catégorisation, la majorité des erreurs sont dues à une mauvaise fusion des deux corpus, notamment à cause du mauvais format/remplissage/paramétrage des balises. Cependant, savoir repérer ces

différents types d'erreurs aide à les résoudre de manière rapide et aisée. Dans le cas de ce projet avec l'entreprise, nous avons plusieurs façons de résoudre les problèmes rencontrés. Nous pouvons aller directement sur le backoffice pour modifier un ou plusieurs modèles directement dans les parties concernés, sur l'interface d'administration pour corriger/ajouter des règles, sur les data pour modifier le corpus de données d'entrées ou en dernier recours, nous pouvons faire appel à l'équipe Technique de l'entreprise en rédigeant un ticket. Cette solution doit être envisagée seulement si le problème rencontré est insoluble si non réglé au niveau du code des outils.

4.2.3 Textes obtenus



Texte n° 1 (323 mots pour 1672 signes)

h2 > Savigny sur Orge, l'évolution des prix de l'immobilier

La ville de Savigny sur Orge connaît une augmentation de 3 % du volume de transaction dans l'ancien depuis mai 2020, avec des prix à la baisse. Les maisons enregistrent une baisse de -4 % et les appartements de -6 %. À Savigny sur Orge, le prix du m² est de 2 400 € environ aujourd'hui, contre 2 600 € en 2018. Les appartements de type T3 bis se sont vendus plus cher en 2019 qu'en 2017 : effectivement, on peut distinguer une hausse de 3,5 %. Ces dernières années, c'est le bien dont le taux a le plus augmenté. À l'inverse, la valeur des appartements de type Loft a diminué (4,5 % moins chers entre ces deux mêmes années).

Concernant le neuf, la tendance est mitigée. En 2017, le prix du m² neuf pour une maison était de 3 100 €, il est actuellement à 2 700 €. On observe une hausse du prix des appartements récemment depuis 2016 : en effet, le prix du m² est passé de 1 600 € à 1 765 € en moyenne depuis 2016, pour une augmentation de 3 %.

h3 > En ce qui concerne le département de l'Essonne

Quant au département de l'Essonne, les tendances sont un peu différentes. En 2018, on peut noter que les appartements et maisons essonniennes s'écoulaient à 1 677 € le m² neuf, et 2 700 € pour l'ancien environ. La tendance est à la hausse pour le prix des appartements (+6,3 %) et pour le prix des maisons (+5 %) depuis un an. La ville du département qui connaît la plus grosse augmentation de prix est Morsang-sur-Orge : +3,7 % entre 2016 et 2020. Les prix ont le plus descendu dans la ville de Grigny : en effet, 6,4 % de baisse.

FIGURE 4.1 – Exemple de texte généré par le système de GAT avec modèles

Le texte ci-dessus est un des textes générés par le système de GAT avec modèles, après les corrections effectuées suite aux erreurs remontées. Nous constatons ainsi que la mise en page indiquée sur l'Admin lors des réglages de paramètres de plan a bien été respectée, et que tous les attributs ont été correctement utilisés. Le texte est grammaticalement correct, et parfaitement lisible. Après vérification, les données d'entrée figurent bien dans le texte, qui répond donc à son objectif informatif.

4.3 Système de GAT avec réseau neuronal

4.3.1 Essais

Il a fallu faire de nombreux essais pour trouver la configuration idéale pour le réseau de neurones. Nous allons donc présenter les différents résultats obtenus avec leur configuration, afin de montrer l'évolution de la qualité des textes générés.

Tests de génération avec une configuration par défaut

En revanche, le neuf peine dus aport prix dump pour less a laugh de les at les appartements entre est a llettements aroun a chuté de même pour les pour ou les periodies des prix des appartements a les prix dumps a luches on test (3 4 150 € ten prix pour les prix desor € les peises ont leptouch et
Versailles suit les tendances des Yvelines. En moyenne, en 2,2 900 € / maison de 2019200. En mai en les appartements et 15m des prix depuis et 2019, les annor cloudences de prix est Fen prix depuis prix approm 90 dden entre chuté de maisons le plus prix depuis que le suids a leg annéesse event pr
Versailles suit my sub a lle13 prix appart prix augm appartements au prix dump prix du m neuf +1,2 %eplanets entre chutaes de même pour les in aujort prix ont lepto, chute mouse depuis quelques années qui chute me (3 800 € en 2019 2019)

TABLE 4.3 – Exemple de phrases générées via textgenrnn avec les paramètres par défaut, avec un entraînement de 50 epoch et un corpus d’un texte

Dans le tableau ci-dessus, les textes générés contiennent des mots français, mais cela représente environ 10 % des textes. Le reste des mots est incompréhensible, même si il est possible de distinguer une certaine structure de phrase.

La ville de Savigny sur Orge connat une augmentation de 3 ‘% du volume de transaction dans l’ancien (+15 %) est perceptible pour la ville de Toulouse, avec des prix mitigés. Effectivement, on remarque une évolution de +4,3 ‘% pour les maisons, et -2,8 % pour les appartements. 3 400 €, c’était le prix
Depuis février 2020, une augmentation du volume de transaction dans l’ancien est visible depuis juin 2020, avec des prix globalement à la baisse. On peut noter que les maisons ont enregistré une baisse de 6,6 % est en ce pour les maisons ont connu une baisse de 4,6 %, le prix du m neuf pour une mai
On peut constater au Mans une diminution du volume de transaction de 10 % dans l’ancien, avec des prix à la baisse. Ainsi, pour les maisons on a un taux de -6,6 %, et de -3 % pour les appartements depuis juin 2020. En 2019, le prix du m au Mans s’élevait à 3 000 € environ, pour 2 600 € à présent. E

TABLE 4.4 – Exemple de phrases générées via textgenrnn avec les paramètres par défaut, avec un entraînement de 50 epoch et un corpus de 10 textes

Pour le tableau ci-dessus, tous les mots présents dans les textes sont français, les structures de phrases sont plus définies que pour le test précédent. On constate une amélioration dûe à la taille du corpus qui est plus important que pour le test précédent. Cependant, malgré une lisibilité accrue, le sens des phrases générées après la première phrase de chacun des textes reste encore assez flou.

Depuis février 2020, une augmentation du volume de transaction dans l'ancien (+15 %) est visible pour la ville de Toulouse, avec des prix mitigés. En effet, on observe une évolution de +4,3 % pour les maisons, et -2,8 % pour les appartements. Le prix du m à Toulouse est en ce moment à 3 600 € envir
Dans la ville de Marseille, on peut observer que le volume de transaction dans l'immobilier ancien est en hausse de 5 %, et ce depuis mars 2020, avec des prix à la hausse : +5 % pour les maisons et +7,9 % pour les appartements. En 2018, le prix du m à Marseille était de 3 000 € environ, pour 3 360
Depuis février 2020, une augmentation du volume de transaction dans l'ancien depuis mai 2020, avec des prix à la baisse. Les maisons enregistrent une baisse de -4 % et les appartements de -6 %. Le prix du m à Savigny sur Orge est présentement à 2 400 € environ, contre 2 600 € en 2018. Les apparteme

TABLE 4.5 – Exemple de phrases générées via textgenrnn avec les paramètres par défaut, avec un entraînement de 100 epoch et un corpus de 30 textes

Pour ce troisième essai, nous constatons que les textes sont lisibles, et cohérents dans les informations fournies : quand nous lisons « à la hausse », juste après nous avons bien des taux positifs. Nous pouvons tout de même repérer quelques répétitions de mots, mais à part cela les textes sont nettement meilleurs que lors des tests précédents.

Depuis février 2020, une augmentation du volume de transaction dans l'ancien (+15 %) est perceptible pour la ville de Toulouse, avec des prix mitigés. En effet, on observe une évolution de +4,3 % pour les maisons, et -2,8 % pour les appartements. Le prix du m à Toulouse est présentement à 3 600 € e
On peut remarquer au Mans une baisse du volume de transaction de 10 % dans l'ancien, avec des prix à la baisse. Ainsi, pour les maisons on a un taux de -6,6 %, et de -3 % pour les appartements depuis juin 2020. En 2019, le prix du m au Mans s'élevait à 3 000 € environ, pour 2 600 € à présent. En 20
La ville de Savigny sur Orge présente une hausse de 3 % du volume de transaction dans l'ancien (+15 %) est perceptible pour la ville de Toulouse, avec des prix mitigés. Effectivement, on constate une évolution de +4,3 % pour les maisons, et -2,8 % pour les appartements. Le prix du m à Toulouse est

TABLE 4.6 – Exemple de phrases générées via textgenrnn avec les paramètres par défaut, avec un entraînement de 200 epoch et un corpus de 30 textes

Avec les textes du tableau précédent, les phrases de ce tableau sont les meilleurs résultats obtenus pour tous les tests effectués. Ils sont certes tronqués en milieu de la phrase à la fin, mais les informations sont présentes, et les textes sont faciles à lire.

Tests de génération avec une configuration personnalisée

Depuis février 2020, une augmentation de -2,7 % depuis 2016, pour une maison était de 4 120 € en 2016 et 2020. Les prix qui baissent par rapport à l'année précédentes années). Concernant le neuf, l'évolution de prix du m ² à Versailles à une chute de -4 % et les appartements et maisons ont +5,7 % de
Dans la ville de Savigny sur Orge est présentement à 2 400 € en moyenne, en 2018, le prix du m ² pour une maison, le prix du m ² neuf était de 3 100 €, il est aujourd'hui de 1 600 € à 1 765 € environ, contre 3 400 € environ à 3 100 € le m ² en 2017 et 2020. Les prix ont le plus augmentation. D'un autr
Depuis février 2020, une augmentation de 3,8 %, avec une baisse de -4 % et les appartements et maisons on a un taux de -4,9 %, et de -3 % pour les appartements et maisons ont +5,7 % de hausse de 5 %, et ce depuis 2016, alors qu'il est actuellement à 2 400 € en 2017 et 2020. Les prix ont le plus aug

TABLE 4.7 – Exemple de phrases générées via textgenrnn avec des paramètres définis, avec un entraînement de 100 epoch et un corpus de 30 textes

Pour le tableau ci-dessus, nous constatons que les textes générés ne possèdent pas une structure correcte de phrase, et que les informations ne sont pas cohérentes. Les répétitions d'informations sont également fréquentes.

On peut constate une diminution de prix est Feucherolles : +9 % entre 2017 et 2020. Les prix ont le plus descendu dans la ville de Toulouse suit le mouvement de la Sarthe, les tendances sont quelque peu différentes. Les appartements, pour les appartements et maisons ont +5,7 % de la valeur ces dern
On peut noter que les appartements de type T2 bis a baissé. Le neuf, et 2 980 € pour l'ancien en moyenne, en 2017, le prix du m ² est de 1 500 €. Le prix du m ² est passé de 3 000 € en 2016 : effectivement, de 6,3 % et 5 %. La ville d'Aix en Provence a enregistrent une chute de -4 % et les appartemen
On peut noter que les prix ont le plus descendu entre ces deux mêmes années, c'est à Aubagne : en effet, on observer que les appartements et maisons on a un taux de -6,6 %, et de -3 % pour les appartement. Les appartements et maisons a augmentation de 10 % dans l'ancien est en adéquation avec les t

TABLE 4.8 – Exemple de phrases générées via textgenrnn avec des paramètres définis, avec un entraînement de 200 epoch et un corpus de 30 textes

Les textes de ce tableau sont certes plus lisibles que lors de l'essai précédent, mais les données ne sont pas cohérentes : un taux positif suit « une diminution de prix ». De plus, certaines parties de phrases n'ont aucun sens.

A l'issue des tests, nous pouvons conclure que la meilleure configuration pour la génération de textes avec un réseau de neurones est celle avec les paramètres d'origine du module, un corpus de 30 textes et un entraînement de 100 epochs (voire 200).

4.4 Conclusion

Les textes générés via système de GAT avec réseau neuronal sont très courts, et donc incomplets. Cependant, les phrases formées sont correctes dans la meilleure des configurations, et les informations données correspondent bien aux données d'entrée. Les textes générés via système de GAT avec modèles sont quant à eux de la bonne longueur, contiennent toutes les informations et sont lisibles. Nous pouvons ainsi déclarer que les textes du système de GAT avec modèles sont les mieux adaptés pour le but donné, cependant leur génération prend beaucoup de temps. Il serait sans doute possible d'obtenir des résultats équivalents en travaillant davantage sur le système de GAT avec réseau neuronal, pour ainsi gagner un temps humain précieux et obtenir des textes valables.

EVALUATION

Sommaire

5.1	Introduction	53
5.2	Type d'évaluation	53
5.2.1	Formulaire d'évaluation	54
5.2.2	Critères d'évaluation	55
5.3	Analyse des résultats des évaluations	56
5.3.1	Présentation des résultats	57
5.4	Que retenir de cette évaluation?	59
5.5	Conclusion	59

5.1 Introduction

Suite à nos expériences, nous avons décidé de faire une évaluation des textes obtenus. En effet, l'évaluation fait partie intégrante du domaine de la GAT, et il est intéressant de comparer les différents résultats obtenus. Nous allons donc dans cette partie, décider d'un système d'évaluation qui réponde aux besoins et contraintes, et également présenter le format d'évaluation choisi ainsi que les résultats obtenus.

5.2 Type d'évaluation

Nous avons décidé d'avoir recours à une évaluation humaine, et non métrique. En effet, nous pensons que l'évaluation humaine est plus pertinente pour ce type de textes, car nous cherchons à observer la structure de la langue ainsi que l'effet produit sur les personnes qui les liront. Les textes que nous avons produits sont à destination de personnes qui n'ont pas de grandes connaissances concernant le domaine immobilier. Il semble alors normal de choisir des personnes avec un niveau de connaissances semblables pour évaluer les textes.

Nous avons ainsi choisi de faire évaluer nos textes par une dizaine de personnes, avec des profils assez différents. Nous avons donc demandé à quelques-uns de nos collègues de stage et à des personnes de notre entourage, d'âge et de parcours différents. Nous espérons atteindre un panel de personnalités assez diverses en faisant cela.

Il s'agira alors d'opérer une évaluation intrinsèque, par rapport à la qualité des textes obtenus, à leur utilité, et à leur véracité.

Comme [Hastie and Belz, 2014] le démontrent, il y aurait trois types d'évaluations pour la GAT :

- NLG components in isolation (composants de GAT isolés);
- end-to-end NLG systems (systèmes de GAT de bout en bout);
- embedded component (systèmes de GAT intégrés);

Nous faisons le choix ici de considérer les textes obtenus dans leur ensemble, cela correspondant à l'évaluation « end-to-end ». En effet, dans le cadre de notre stage, nous avons observé qu'il était primordial d'évaluer la qualité des textes dans leur ensemble, car c'est de cette manière qu'ils sont ensuite livrés aux clients.

5.2.1 Formulaire d'évaluation

Il s'agit ensuite de déterminer le nombre de textes à faire évaluer pour chaque expérience. Il serait judicieux de choisir des textes aux données différentes. Comme nous avons produit des données pour 6 villes, nous pouvons donc choisir 5 textes correspondant à 5 de ces villes. Il y aura donc 10 textes au total, 5 pour chaque expérience menée, choisis de manière aléatoire. Une fois les textes sélectionnés, il s'agit de choisir la meilleure manière de les présenter aux personnes qui les évalueront, ainsi que de penser à comment organiser l'évaluation.

Nous avons créé un document au format Excel composés de plusieurs pages, dont chacune correspondant à un texte différent. Pour chaque page, nous pouvons trouver le texte en question, les données d'entrées nécessaires à sa génération, ainsi que le formulaire d'évaluation.

Le formulaire d'évaluation sera sous la forme d'un tableau, contenant les 3 critères dont nous reparlerons dans la partie suivante. Les évaluateurs pourront donc évaluer chacun des paragraphes des textes présentés, s'ils en contiennent, ainsi que le texte dans sa globalité. Egalement, il leur est possible d'écrire des commentaires si le besoin s'en fait ressentir.

Les textes 1 à 5 sont de textes longs, obtenus avec le système de modèles, et les textes 6 à 10 sont des textes courts, obtenus avec le second système. La notation sera donc légèrement différente, car il ne sera pas possible d'effectuer une notation par paragraphes pour les textes 6 à 10. Les premiers textes seront donc notés avec le tableau visible dans la figure ci-dessous, et une version simple avec seulement la ligne "texte entier" sera fournie pour les textes courts.

Les critères Lisibilité, Véracité et Utilité pourront être notés avec une échelle de Likert, allant de 1 à 5, comme celle utilisée par [Belz and Reiter, 2006]. Sur cette échelle, 1 renvoie à "très mauvais" et 5 à "très bon", 3 étant ainsi la note neutre.

Nous faisons l'hypothèse que les résultats de l'évaluation seront meilleurs pour les textes obtenus avec le système de GAT avec modèles, du fait de leur longueur et des multiples vérifications qualitatives effectuées au cours du projet.

Partie à évaluer	Lisibilité	Véracité	Utilité	Commentaire
Paragraphe 1 (données globales)				
Paragraphe 2 (maisons et appartements neufs)				
Paragraphe 3 (données département)				
Texte entier				

FIGURE 5.1 – Formulaire d'évaluation fourni aux évaluateurs pour chacun des textes

5.2.2 Critères d'évaluation

Après réflexion, nous avons décidé de reprendre les critères cités par [Reiter, 2019] :

- Lisibilité (Linguistic quality, fluency, clarity, or readability);
- Véracité (Accuracy, correctness);
- Utilité (Utility, usefulness or helpfulness);

Nous allons expliciter chacun de ces critères dans les sous-parties suivantes.

Lisibilité

*Is the text easy to read and understand?*¹

Nous cherchons à savoir avec ce critère, si les textes sont bien écrits, peu importe si le contenu est correct. Il est nécessaire que le texte soit compréhensible à la lecture de celui-ci, sans fautes, et que les phrases soient correctement formées. Nous pourrions également déterminer si ce critère est rempli grâce à la mesure de la qualité linguistique que nous pourrions en faire, ou de la vitesse de lecture d'un être lambda sur ce texte. Il serait alors nécessaire de s'assurer que tous les évaluateurs lisent de la même manière, avec la même application. Cependant, ici, nous allons simplement demander aux évaluateurs de noter le critère d'après leur notion intrinsèque de qualité linguistique.

Véracité

*Is everything in the text true and derivable from the input data?*²

Pour la véracité, il est nécessaire de déterminer si le texte final dérive bien des données d'entrée insérées dans le système. Ce critère dépend grandement de la notion de "précision" : tout ce qui se trouve dans le texte doit venir des données entrées dans

1. "Est-ce que le texte est simple à lire et à comprendre?"

2. "Est ce que l'entièreté du texte est vraie et provenant des données d'entrée?"

le système. Cependant, la notion de "rappel" ne joue pas sur ce critère, car toutes les données d'entrée communiquées ne doivent pas figurer dans un texte. Dans notre cas, un texte parlant de Versailles ne doit pas inclure de données concernant Le Mans ou Toulouse.

Un autre aspect de ce critère à prendre en compte est le fait que le texte peut être vrai dans son ensemble, mais si des données vraies figurent dans le texte et ne viennent pas du corpus de données d'entrées, il n'est pas possible de valider ce texte comme étant vrai. C'est pourquoi, pour les besoins de notre évaluation, nous avons inclus les données d'entrée de chaque texte dans le formulaire. Ainsi, l'évaluateur peut vérifier lui-même si le texte ne contient que des données présentes dans le corpus.

Utilité

*Does the text help a user do a task, or otherwise achieve its communicative goal?*³

L'utilité mesure à quel point un texte est utile : c'est-à-dire, est-ce qu'il remplit correctement son but informatif? Ce critère est bien évidemment influencé par les deux autres critères. Néanmoins, nous pouvons également déterminer l'utilité des textes par rapport à la pertinence des données utilisées. Il est possible aussi de mesurer l'utilité par rapport à la prise de décision provoquée par des textes générés [Portet et al., 2009], ou le changement d'attitude [Reiter et al., 2003]. Cependant, la manière la plus simple de déterminer l'utilité d'un texte est encore de demander l'avis de réelles personnes, avec une échelle de Likert, d'après [Reiter, 2019].

5.3 Analyse des résultats des évaluations

A l'issue de l'évaluation, nous avons obtenu 8 formulaires remplis de notes et de commentaires. Nous n'avons pas pu avoir plus de participation, en raison du contexte sanitaire actuel lors de la rédaction de ce mémoire, malheureusement. Cependant, nous pouvons déjà distinguer quelques tendances se dégager des résultats obtenus.

Il a semblé y avoir quelques erreurs d'incompréhension, que nous avons pu remarquer suite à la première lecture des résultats obtenus (suite à des questions/-réponses, certaines notations ont pu être modifiées par les participants concernés) :

- le critère de l'utilité a été moyennement compris par certains participants. Ceux-ci ne savaient pas exactement à quoi cela faisait référence. Nous avons donc ré-expliqué plus profondément les subtilités de ce critère ;
- la véracité n'a pas toujours été notée de manière correcte : en effet, certains textes ont obtenu des notes parfaites alors qu'ils ne contenaient pas toutes les données d'entrée par exemple. Le critère a donc été expliqué plus en détail

Nous avons cependant pu remarquer une bonne compréhension du critère de la lisibilité, ainsi qu'une certaine intransigeance de la part de nos évaluateurs en ce qui concernait les possibles fautes d'orthographe, ainsi que pour des tournures de phrases moyennes. Nous avons donc pu recueillir 8 évaluations complètes et très

³. "Est-ce que le texte peut aider un lecteur à accomplir une tâche, ou est-ce qu'il accomplit un but communicatif?"

intéressantes pour notre projet.

Nous avons regroupé les notations, pour avoir un meilleur aperçu global de nos résultats. Également, nous avons opéré quelques opérations, notamment avec les calculs de la moyenne et de la médiane des notes de chaque texte : pour les textes 1 à 5, seules les notes concernant la globalité du texte en question ont été utilisées. Nous avons choisi d'effectuer le calcul des moyennes, car cela peut faire ressortir une certaine tendance, entre les textes obtenus avec le premier système et les textes obtenus avec le second système. En effet, la moyenne est le calcul classique effectué lorsque l'on rencontre une liste de nombres. Cependant, celle-ci sera influencée par les valeurs extrêmes de cette liste. Ainsi, si la liste contient des valeurs extrêmes, la moyenne ne sera pas vraiment représentative. C'est pourquoi nous avons également décidé d'inclure la médiane dans nos calculs, afin d'avoir une meilleure représentation de nos notes.

5.3.1 Présentation des résultats

Texte	Lisibilité	Véracité	Utilité
1	4	4,875	4,875
2	3,875	5	4,875
3	4,5	4,875	4,875
4	4,5	5	4,875
5	4,375	5	4,875
6	3,625	2,75	2,75
7	3,125	2,75	3,125
8	3,375	2,375	3
9	3,375	2,625	3
10	1,625	1,625	1,75

TABLE 5.1 – Moyenne des résultats de l'évaluation sur les 10 textes

Texte	Lisibilité	Véracité	Utilité
1	4	5	5
2	4	5	5
3	5	5	5
4	5	5	5
5	4	5	5
6	4	3	3
7	3	3	3
8	4	3	3
9	4	3	3
10	1,5	2	1,5

TABLE 5.2 – Médiane des résultats de l'évaluation sur les 10 textes

	Lisibilité	Véracité	Utilité
Moyenne	3,6375	3,6875	3,8
Médiane	3,85	3,9	3,85

TABLE 5.3 – Moyenne et médiane des notes attribuées aux trois critères pour l'ensemble des textes

Dans l'ensemble, nous pouvons constater que les résultats sont au-dessus de la moyenne. Le critère de l'utilité semble avoir été le mieux noté, avec une moyenne de 3,8. Cela peut s'expliquer par le fait que les textes 6 à 10 n'ont pas été vraiment impactés par la notation de ce critère au cours de leur notation, malgré leur moyenne basse. En effet, ces textes sont très courts, ils ne contiennent donc pas toutes les données d'entrée, ce qui fait baisser d'office la notation de la véracité. De plus, les phrases sont moins bien construites, faisant également baisser la notation de la lisibilité. L'utilité en revanche n'a pas été vraiment impactée, du fait que les phrases contiennent tout de même des informations (vraies ou non, cela n'est pas jugé dans la notation de ce critère) qui peuvent être utiles dans un certain contexte.

Egalement, nous pouvons noter un écart significatif dans la notation des textes : les textes 1 à 5 bénéficient d'une meilleure moyenne que les textes suivants. Pour observer cela plus clairement, nous avons calculé la moyenne et la médiane pour ces deux groupes de textes.

	Lisibilité	Véracité	Utilité
Moyenne	4,25	4,95	4,875
Médiane	4,4	5	5

TABLE 5.4 – Moyenne et médiane des notes attribuées aux textes 1 à 5

	Lisibilité	Véracité	Utilité
Moyenne	3,025	2,425	2,725
Médiane	3,3	2,8	2,7

TABLE 5.5 – Moyenne et médiane des notes attribuées aux textes 6 à 10

Comme nous pouvons le voir dans les tableaux ci-dessus, une tendance se dégage clairement entre ces deux groupes de textes. Les textes issus de la génération avec réseau de neurones ont été moins bien notés que ceux issus de la génération avec modèles. Le texte 10 a été particulièrement mal noté : en effet, il s'agissait d'un exemple de texte possédant les données d'entrées de 2 villes différentes, résultant donc en un texte difficilement compréhensible.

5.4 Que retenir de cette évaluation ?

Dans l'ensemble, les notes obtenues pour l'évaluation sont bonnes et au dessus de la moyenne de 2,5. Les textes 1 à 5 ont été particulièrement bien notés. Cependant, le critère de la lisibilité a été le moins bien noté dans ce cas, alors que pour les textes 6 à 10 il s'agit du critère le mieux noté. Nous avons remarqué, via les commentaires de nos évaluateurs, une certaine intransigeance en ce qui concerne l'orthographe ou les tournures de phrases des textes 1 à 5, résultant donc en une moyenne plus basse pour ce critère. Alors qu'au contraire, la lisibilité bénéficie de la meilleure moyenne pour les textes suivants. Nous pensons que les évaluateurs ont été plus conciliants pour ces textes, chose dont les autres critères n'ont pas pu bénéficier car il n'est pas possible de mieux noter la véracité, au vu des données incomplètes, voire fausses dans certains textes. Le critère de l'utilité également a subi une baisse de moyenne, notamment à cause de la longueur des textes, jugée insuffisante par nos évaluateurs pour pouvoir vraiment informer les personnes lisant ces textes.

Nous avons remarqué lors de la notation une plus grande intransigeance de la part des personnes qui ne travaillent pas chez Labsense. Les notes étaient plus sévères, notamment au niveau de la lisibilité. Il est possible que l'expérience de la génération automatique de textes ait influencé le niveau de conciliation de ces personnes.

Nous pourrions améliorer cette évaluation en effectuant quelques changements : rédiger une meilleure description des critères car il y a eu quelques problèmes de compréhension, ou faire évaluer les textes par de potentiels utilisateurs de ces textes. En effet, le critère de l'utilité serait mieux représentatif si les personnes qui évaluent les textes ont actuellement besoin de ces informations. De plus, nous pourrions également tester nos textes en passant cette fois par une évaluation métrique.

5.5 Conclusion

Dans ce chapitre, nous avons présenté notre démarche pour l'évaluation menée, avec les formulaires utilisés, les critères que nous avons choisis, et les résultats que nous avons obtenus. Nous avons ensuite interprété ces résultats et proposé des améliorations qui pourraient rendre cette évaluation plus efficace.

CONCLUSION GÉNÉRALE

Dans ce mémoire, nous avons décrit le fonctionnement de deux systèmes de GAT, l'un avec des modèles, l'autre avec un réseau neuronal. Nous avons défini un état de l'art concernant le domaine des systèmes de GAT, et l'avons relié à notre problématique de recherche autour des questions que nous avons posé lors de l'introduction de ce mémoire. Nous avons confronté ces deux systèmes dans le but de définir leurs spécificités, mais également leurs défauts quant au projet de génération de textes immobiliers que nous avons effectué.

Notre recherche nous a permis de définir quelques points fondamentaux sur le domaine de la génération automatisée de textes :

D'une part, il est important de choisir le type de système de GAT avec attention, en prenant en compte les données d'entrée, le type de rendu souhaité, les capacités en travail humain que l'on souhaite accorder au projet et le domaine concerné.

Egalement, nous avons compris l'importance du format des données d'entrées, ainsi que le processus de fusion de ces données avec le corpus de modèles dans le cas du système de GAT avec modèles. En ce qui concerne le système de GAT avec réseau neuronal, nous avons fait l'expérience des réglages du réseau de neurones, qui sont essentiels au bon fonctionnement du système.

Nous avons proposé et effectué une évaluation humaine reposant sur les textes obtenus avec ces deux systèmes de GAT. Nous avons recueilli les notes et commentaires de nos évaluateurs, et avons analysé ces mêmes résultats dans le but de repérer des points d'amélioration de nos systèmes et de notre processus d'évaluation, et nous avons également proposé des façons d'améliorer ce processus.

Perspectives

Si nous avons l'opportunité de poursuivre ce travail de recherche, nous nous concentrerions sur d'autres types de systèmes de GAT, dans le but de continuer notre travail de comparaison. Nous approfondirions également ce même projet, en améliorant les points que nous avons mentionné précédemment. Il serait notamment intéressant d'effectuer une évaluation métrique avec d'autres systèmes de GAT, afin de voir la différence avec l'évaluation humaine.

BIBLIOGRAPHIE

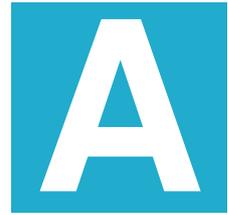
- [Banaee et al., 2013] Banaee, H., Ahmed, M. U., and Loutfi, A. (2013). Towards nlg for physiological data monitoring with body area networks. In *14th European Workshop on Natural Language Generation, Sofia, Bulgaria, August 8-9, 2013*, pages 193–197. – Cité page 17.
- [Bartoli et al., 2016] Bartoli, A., De Lorenzo, A., Medvet, E., and Tarlao, F. (2016). Your paper has been accepted, rejected, or whatever: Automatic generation of scientific paper reviews. In *International Conference on Availability, Reliability, and Security*, pages 19–28. Springer. – Cité page 15.
- [Belz and Reiter, 2006] Belz, A. and Reiter, E. (2006). Comparing automatic and human evaluation of nlg systems. In *11th Conference of the European Chapter of the Association for Computational Linguistics*. – Cité pages 21 et 54.
- [Binsted et al., 1997] Binsted, K., Pain, H., and Ritchie, G. D. (1997). Children’s evaluation of computer-generated punning riddles. *Pragmatics & Cognition*, 5(2):305–354. – Cité page 16.
- [Bringsjord and Ferrucci, 1999] Bringsjord, S. and Ferrucci, D. (1999). *Artificial intelligence and literary creativity: Inside the mind of brutus, a storytelling machine*. Psychology Press. – Cité page 16.
- [Brown et al., 2005] Brown, J., Frishkoff, G., and Eskenazi, M. (2005). Automatic question generation for vocabulary assessment. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 819–826. – Cité page 17.
- [Clarke and Lapata, 2010] Clarke, J. and Lapata, M. (2010). Discourse constraints for document compression. *Computational Linguistics*, 36(3):411–441. – Cité pages 15 et 16.
- [Dale and Mellish, 1998] Dale, R. and Mellish, C. (1998). Towards evaluation in natural language generation. In *In Proceedings of First International Conference on Language Resources and Evaluation*. – Cité page 20.
- [Dušek et al., 2018] Dušek, O., Novikova, J., and Rieser, V. (2018). Findings of the e2e nlg challenge. *arXiv preprint arXiv:1810.01170*. – Cité pages 5 et 20.
- [Gatt and Krahmer, 2018] Gatt, A. and Krahmer, E. (2018). Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research* 61 (2018) 65-1, pages 65–170. – Cité pages 5, 15, 17, 18, 19 et 21.
- [Hastie and Belz, 2014] Hastie, H. F. and Belz, A. (2014). A comparative evaluation methodology for nlg in interactive systems. In *LREC*, pages 4004–4011. – Cité page 54.

- [Hüske-Kraus, 2003] Hüske-Kraus, D. (2003). Suregen-2: A shell system for the generation of clinical documents. In *Demonstrations*. – Cité page 17.
- [Jones and Galliers, 1995] Jones, K. S. and Galliers, J. R. (1995). *Evaluating natural language processing systems: An analysis and review*, volume 1083. Springer Science & Business Media. – Cité page 21.
- [Kulkarni et al., 2013] Kulkarni, G., Premraj, V., Ordonez, V., Dhar, S., Li, S., Choi, Y., Berg, A. C., and Berg, T. L. (2013). Babytalk: Understanding and generating simple image descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12):2891–2903. – Cité pages 5 et 17.
- [Langkilde, 2002] Langkilde, I. (2002). An empirical verification of coverage and correctness for a general-purpose sentence generator. In *Proceedings of the international natural language generation conference*, pages 17–24. – Cité page 21.
- [Langkilde-Geary and Knight, 2002] Langkilde-Geary, I. and Knight, K. (2002). Halogen statistical sentence generator. In *Proceedings of the ACL-02 Demonstrations Session*, pages 102–103. Citeseer. – Cité page 19.
- [Leppänen et al., 2017] Leppänen, L., Munezero, M., Granroth-Wilding, M., and Toivonen, H. (2017). Data-driven news generation for automated journalism. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 188–197. – Cité page 17.
- [Lin and Hovy, 2003] Lin, C.-Y. and Hovy, E. (2003). Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 150–157. – Cité page 21.
- [Nesterenko, 2016] Nesterenko, L. (2016). Building a system for stock news generation in russian. In *Proceedings of the 2nd International Workshop on Natural Language Generation and the Semantic Web (WebNLG 2016)*, pages 37–40. – Cité page 17.
- [Papangelis et al., 2019] Papangelis, A., Wang, Y.-C., Molino, P., and Tur, G. (2019). Collaborative multi-agent dialogue model training via reinforcement learning. *arXiv preprint arXiv:1907.05507*. – Cité page 16.
- [Papineni et al., 2002] Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318. – Cité page 21.
- [Portet et al., 2009] Portet, F., Reiter, E., Gatt, A., Hunter, J., Sripada, S., Freer, Y., and Sykes, C. (2009). Automatic generation of textual summaries from neonatal intensive care data. *Artificial Intelligence*, 173(7-8):789–816. – Cité page 56.
- [Reiter, 2018] Reiter, E. (2018). Hallucination in neural nlg. <https://ehudreiter.com/2018/11/12/hallucination-in-neural-nlg/>. – Cité page 19.
- [Reiter, 2019] Reiter, E. (2019). Accuracy, fluency, and utility. <https://ehudreiter.com/2019/10/08/accuracy-fluency-and-utility/>. – Cité pages 20, 21, 55 et 56.

- [Reiter and Dale, 2000] Reiter, E. and Dale, R. (2000). Building natural language generation systems. *Studies in Natural Language Processing*. – Cité pages 15 et 18.
- [Reiter et al., 2003] Reiter, E., Robertson, R., and Osman, L. M. (2003). Lessons from a failure: Generating tailored smoking cessation letters. *Artificial Intelligence*, 144(1-2):41–58. – Cité pages 21 et 56.
- [Reiter et al., 2005] Reiter, E., Sripada, S., Hunter, J., Yu, J., and Davy, I. (2005). Choosing words in computer-generated weather forecasts. *Artificial Intelligence*, 167(1-2):137–169. – Cité page 16.
- [Siddharthan and MacDonald, 2016] Siddharthan, A. and MacDonald, I. (2016). Summarising news stories for children. In *Proceedings of the 9th International Conference on Natural Language Generation (INLG 2016), Edinburgh, Scotland*. ACL. – Cité pages 16 et 17.
- [van der Lee et al., 2017] van der Lee, C., Krahmer, E., and Wubben, S. (2017). Pass: A dutch data-to-text system for soccer, targeted towards specific audiences. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 95–104. – Cité page 17.
- [Weizenbaum, 1966] Weizenbaum, J. (1966). Eliza—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1):36–45. – Cité page 16.
- [Woolf, 2017] Woolf, M. (2017). textgenrnn. <http://github.com/minimaxir/textgenrnn>. – Cité page 41.
- [Yang et al., 2004] Yang, Z. R., Everson, R., and Yin, H. (2004). *Intelligent Data Engineering and Automated Learning-IDEAL 2004: 5th International Conference, Exeter, UK, August 25-27, 2004, Proceedings*, volume 3177. Springer. – Cité page 16.
- [Yngve, 1961] Yngve, V. H. (1961). *Random generation of English sentences*. Massachusetts Inst. of Technology. – Cité page 16.
- [Zock and Sabah, 1992] Zock, M. and Sabah, G. (1992). La génération automatique de textes : trente ans déjà, ou presque. *Langages*, 26 année, n°106, pages 8–35. – Cité page 16.

Troisième partie

Annexes



DOCUMENTATION

A.1 Langages et bibliothèques

Ce mémoire a été mis en page avec LaTeX, via l'éditeur en ligne **Overleaf**.

Nom	Version
Libre Office	6.2.5.2
Python	3.5.2

TABLE A.1 – Langages et logiciels utilisés dans le cadre de ce mémoire

Nous avons également utilisé le module Python **textgenrnn**.

A.2 Corpus

La table A.2 est un extrait d'un des corpus utilisés pendant la partie du projet concernant le système de GAT avec réseau neuronal. Les textes visibles dans cette partie proviennent des textes obtenus avec le réseau de GAT avec modèles.

A Versailles, le volume de transaction dans l'ancien est en hausse de 11 % depuis janvier 2020, avec des prix globalement orientés à la hausse : +3,7 % pour les maisons, et +2,65 % pour les appartements par rapport à l'année précédente. En 2019, le prix du m² à Versailles s'élevait à 3 200 € en moyenne, pour 3 500 € aujourd'hui. Les appartements type T3 se sont vendus 4 % plus chers en 2019 qu'en 2018. C'est le type de bien qui a pris le plus de valeur ces dernières années. A l'inverse, la valeur des T6 a chuté (2,5 % moins chers entre ces deux mêmes années). En revanche, le neuf peine davantage et subit une baisse des prix depuis quelques années. Pour une maison, le prix du m² neuf était de 4 150 € en 2016, alors qu'il est aujourd'hui de 3 900 €. Il en va de même pour les appartements, qui subissent une chute de 5,7 % depuis 2019 (3 800 € / m² en 2019 pour 3 554 € aujourd'hui). Versailles suit les tendances des Yvelines. En moyenne, en 2019, les appartements et maisons yvelinoises se vendaient à 3 856 € du m² neuf, ou encore 3 741 € dans l'ancien. Tandis que le prix des appartements a légèrement augmenté (+1,7 %), les maisons ont subi une évolution différente et leur prix a chuté de 2,2 % en un an. La ville du département qui connaît la plus grosse hausse de prix est Feucherolles : +4,2 % entre 2019 et 2020. A l'inverse, c'est à Trappes que les prix ont le plus chuté entre ces périodes (-5,4 %).

La ville de Savigny sur Orge connaît une augmentation de 3 % du volume de transaction dans l'ancien depuis mai 2020, avec des prix à la baisse. Les maisons enregistrent une chute de -4 % et les appartements de -6 %. À Savigny sur Orge, le prix du m² est de 2 400 € en moyenne aujourd'hui, contre 2 600 € en 2018. Les appartements de type T3 bis se sont vendus plus cher en 2019 qu'en 2017 : en effet, on peut percevoir une hausse de 3,5 %. Parmi les types de biens qui ont pris de la valeur ces dernières années, c'est celui qui a eu la plus forte hausse. D'un autre côté, avec une baisse de 4,5 %, la valeur des appartements de type Loft a baissé. Pour ce qui est du neuf, l'évolution est mitigée. En 2017, le prix du m² neuf pour une maison était de 3 100 €, il est actuellement à 2 700 €. Le prix des appartements est de son côté en pleine augmentation depuis 2016, avec une hausse de 3 %, avec un prix au m² qui est passé de 1 600 € à 1 765 € environ. Savigny sur Orge ne suit pas la tendance de son département, l'Essonne. En moyenne, en 2018, les appartements et maisons essonniennes s'écoulaient à 1 677 € du m² neuf, ou encore 2 700 € dans l'ancien. La tendance est à la hausse pour le prix des appartements (+6,3 %) et pour le prix des maisons (+5 %) depuis un an. La ville de Morsang-sur-Orge a enregistré la plus importante hausse de prix du département avec une montée de 3,7 % entre 2016 et 2020. Les prix ont le plus descendu dans la ville de Grigny : effectivement, 6,4 % de baisse.

À Lille, une baisse de 1,5 % du volume de transaction dans l'ancien est perceptible depuis avril 2020, avec des prix à la baisse. On peut noter que les maisons ont connu une baisse de 4,9 % et les appartements de 6 %. À Lille, le prix du m² est de 3 200 € environ aujourd'hui, contre 3 400 € en 2019. En 2018, les appartements de type T2 se sont commercialisés 10 % plus cher qu'en 2016. C'est le type de bien qui a pris le plus de valeur ces dernières années. Toutefois, les appartements de type T1 bis ont enregistré une diminution de leur valeur (19 % moins chers entre ces deux mêmes années). Pour les appartements et maisons neuves, les prix ont progressé de façon hétérogène. En 2016, le prix du m² neuf pour une maison était de 4 000 €, il est actuellement à 4 300 €. Le prix des appartements a chuté depuis 2017 : effectivement, on constate une baisse de -10 % de la valeur, avec 2 500 € le m² en 2017 et 2 100 € aujourd'hui. Lille ne suit pas la tendance de son département, le Nord. En moyenne, en 2017, les appartements et maisons nordistes se vendaient à 3 000 € du m² neuf, ou encore 2 670 € dans l'ancien. Le prix des maisons a diminué de 3,7 % en un an, a contrario des appartements dont le prix a augmenté de 5,7 %. La ville du département qui connaît la plus grosse augmentation de prix est Cambrai : +4,5 % entre 2017 et 2020. Les prix ont le plus chuté dans la ville de Valenciennes : en effet, 10 % de baisse.

TABLE A.2 – Extrait d'un des corpus utilisé pour la génération avec réseau neuronal

A.3 Evaluation

La figure A.1 est un extrait des formulaires d'évaluation décrits dans le chapitre 4. Il y a un formulaire de ce type pour chaque texte, chacun étant contenu dans un onglet d'un Google Sheet. Ils se composent en 3 parties : le texte à évaluer, les données d'entrée et le formulaire de notation accompagné d'un rappel de l'échelle de Likert. Il y a également un onglet "Introduction", dans lequel le projet ainsi que les critères d'évaluation sont expliqués de manière précise. Un onglet "commentaires" a également été mis à disposition des évaluateurs, dans le cas où il y a des commentaires sur l'ensemble de l'évaluation.

3 en lisibilité car la première phrase n'est pas claire, on a l'impression qu'il y a une erreur entre les chiffres
trop grand nombre de données de type différent on se perd entre les taux des volumes/des prix + on parle pas de tous les types d'appartements mais on sait pas pourquoi on parle que de ceux là ?
on s'arrête sur le mot "ancien" dans la première phrase, il faut lire la suite pour en comprendre pleinement le sens
"Lisible cependant quand il y a un chiffre je pense qu'il faudrait ajouter de nouveau ce qu'il indique
Définir certains termes comme ancien car c'est flou si on ne s'y connaît pas en immobilier je pense, ou en tout cas dire quelle période ça englobe.
Certaines tournures de phrases ne sont pas très naturelles. ex : celle qui commence par : "2600 euros, c'était..."
Le texte me semble utile et plutôt clair si l'on veut se renseigner sur le sujet.
Texte compréhensible mais avec quelques phrases peu naturelles.
Il manque bcp de mots de liaisons pour rendre le texte plus léger, fluide et donc lisible
général, se lit tout seul, phrases courtes et claires, construites simplement
le dernier paragraphe se détache des autres
Je rajouterais toujours ce qu'indiquent les taux +4,3% et -2,8%, histoire de faire un rappel et rendre la lecture plus fluide.
On ne sait pas ce que signifie ""m"" dans la seconde phrase.
Il manque énormément d'informations.
Très clair avec mots de liaisons bien adaptés. Lecture fluide
Lecture parfois peu fluide à cause du manque d'averbes + certaines phrases mal construites.
pas de connecteurs entre les phrases : aspect "robotique"-> donne l'impression d'une liste de courses ou de notes de cours, au sens où les infos sont un peu balancées sans arrondir les angles. Incompréhension de la mention de Savigny sur Orge dans un texte portant sur Lille + infos fausses (old proprety rate, now date - with month, old house rate - now)
Pas de début de phrase. Toutes les données ne sont pas présentes. Le m remplace le m ² ce qui peut corrompre la lecture. Texte coupé au milieu d'une phrase
Il manque du contexte dans les phrases.
Le taux de -4% annoncé n'est pas correct.
Les informations ne correspondent pas à ce dont on parle et il en manque énormément.
Il manque un verbe dans la première phrase. Les phrases sont mal construites et la dernière est incomplète. Enfin, il n'y a pratiquement pas d'informations.
là les taux sont bien reliés on comprend que c'est la hausse de prix
1ere phrase bien écrite mais trop longue dommage : trop d'informations ce qui peut rendre la lecture fastidieuse
Toutes les données ne sont pas présentes. Le m remplace le m ² ce qui peut corrompre la lecture. Texte coupé au milieu d'une phrase
Phrase pas terminée.
Il manque énormément d'informations
Dernière phrase incomplète + trop peu d'informations.

TABLE A.3 – Extrait des commentaires recueillis lors de l'évaluation

Texte généré				
	Evolution des prix de l'immobilier à Versailles			
	Depuis mars 2020, on assiste à Versailles à une baisse de 3,67 % du volume de transaction dans l'ancien, avec des prix qui augmentent par rapport à l'année précédente : les maisons ont +5,7 % de hausse, et les appartements +2 %, 2 600 €, c'était le prix du m ² à Versailles en 2019 : il est de 2 900 € désormais. Les appartements de type T1 se sont vendus 7 % plus cher en 2018 qu'en 2016. Ces précédentes années, c'est le bien dont le taux a le plus augmenté. Cependant, les appartements de type T3 ont enregistré une baisse de leur valeur (2,87 % moins chers entre ces deux mêmes années).			
	Pour les appartements et maisons neuves, les prix ont progressé de façon hétérogène. Pour une maison, le prix du m ² neuf était de 3 200 € en 2017, alors qu'il est aujourd'hui de 3 500 €. Les appartements, quant à eux, subissent une chute de -2,7 % depuis 2015 (3 000 € / m ² en 2015 pour 2 900 € actuellement).			
	En ce qui concerne le département des Yvelines			
	Verailles ne suit pas la tendance de son département, les Yvelines. En 2016, on peut noter que les appartements et maisons yvelinoises s'écoulaient à 4 400 € le m ² neuf, et 4 200 € pour l'ancien environ. Concernant le prix des appartements et des maisons, on observe une diminution depuis l'an dernier, avec des chutes de, respectivement, 5,5 % et 3 %. La ville du département qui connaît la plus forte hausse de prix est Faucherolles : +9 % entre 2016 et 2020. Inversement, c'est à Trappes que les prix ont le plus descendu entre ces périodes (-5 %).			
Données d'entrée				
Old property rate	-3,67 %			
Old house rate - now	2 %			
Old house rate - now	5,7 %			
Now date -with month	mars 2020			
Price m ² - comparison year	2 600 €			
Price m ² - now m ²	2 900 €			
Comparison - Type of Apartment	2016			
Comparison - Type of apartment (best year)	2018 T3			
Best type of apartment - now	2019 T1			
Best type of apartment - rate	7 %			
Worst type of apartment - now	2018 T3			
Worst type of apartment - rate	-2,87 %			
Comparison - New House Price	2017			
Formulaire d'évaluation				
Partie à évaluer	Libellé	Véracité	Utilité	Commentaire
Paragraphe 1 (données globales)				
Paragraphe 2 (maisons et appartements neufs)				
Paragraphe 3 (données département)				
Texte entier				
Comment évaluer ?				
1	2	3	4	5
vraiment mauvais	plutôt mauvais	neutre	plutôt bon	vraiment bon

FIGURE A.1 – Formulaire d'évaluation fourni aux évaluateurs pour chacun des textes

EXTRAITS DE SCRIPTS

Comme mentionné dans le chapitre 3, nous avons produit 3 scripts en langage Python dans le but de tester un système de GAT utilisant un réseau de neurones. Il y a donc 3 scripts nécessaires pour produire des textes. Le premier script contient la configuration du réseau de neurones utilisé dans le système, permettant ainsi d'indiquer le nombre de couches de neurones, le nombre de neurones dans chaque couche, etc. Ensuite, il y a un script permettant de lancer l'entraînement du modèle utilisé pour la génération, et enfin, le script permettant de générer les textes avec le modèle obtenu pendant l'entraînement.

B.1 Script contenant la configuration du réseau de neurones

```
1  model_cfg = {
2      'word_level': False,
3      'rnn_size': 128,
4      'rnn_layers': 4,
5      'rnn_bidirectional': True,
6      'max_length': 8,
7      # 'max_words': 100000,
8      "weights_path": "./outputs/"
9  }
10
11  train_cfg = {
12      'line_delimited': True,
13      'num_epochs': 100,
14      'gen_epochs': 1,
15      'train_size': 0.9,
16      'dropout': 0.0,
17      'validation': True,
18      'is_csv': False,
19      "weights_path": "./outputs/"
20  }
21
22  file_name = "corpus_30textes.txt"
23  model_name = "corpus_30txt"
24
```

FIGURE B.1 – Configuration du système (*config.py*)

B.2 Script pour l'entraînement du modèle

```
1     from textgenrnn import textgenrnn
2     from config import *
3
4
5     textgen = textgenrnn(name="./outputs/" + model_name)
6
7     train_function = textgen.train_from_file if train_cfg['line_delimited']
8     else textgen.train_from_targettext_file
9
10    train_function(
11        file_path="./datasets/" + file_name,
12        new_model=True,
13        num_epochs=train_cfg['num_epochs'],
14        gen_epochs=train_cfg['gen_epochs'],
15        batch_size=1024,
16        train_size=train_cfg['train_size'],
17        dropout=train_cfg['dropout'],
18        validation=train_cfg['validation'],
19        is_csv=train_cfg['is_csv'],
20        rnn_layers=model_cfg['rnn_layers'],
21        rnn_size=model_cfg['rnn_size'],
22        rnn_bidirectional=model_cfg['rnn_bidirectional'],
23        max_length=model_cfg['max_length'],
24        dim_embeddings=100,
25        word_level=model_cfg['word_level'])
26
```

FIGURE B.2 – Script d'entraînement du système (*train.py*)

B.3 Script permettant de lancer la génération des textes

```
1     from textgenrnn import textgenrnn
2     from config import *
3     from datetime import datetime
4
5     model_path = "./outputs/" + model_name
6
7     textgen = textgenrnn(weights_path=model_path + '_weights.hdf5',
8                          vocab_path=model_path + '_vocab.json',
9                          config_path=model_path + '_config.json')
10
11    temperature = [1.0, 0.5, 0.2, 0.2]
12    prefix = None
13
14    if train_cfg['line_delimited']:
15        n = 1000
16        max_gen_length = 600 if model_cfg['word_level'] else 300
17    else:
18        n = 1
19        max_gen_length = 600 if model_cfg['word_level'] else 300
20
21    timestring = datetime.now().strftime('%Y%m%d_%H%M%S')
22    gen_file = '{}_gentext_{}.txt'.format(model_name, timestring)
23
24    textgen.generate_to_file("./outputs/" + gen_file,
25                             temperature=temperature,
26                             prefix=prefix,
27                             n=n,
28                             max_gen_length=max_gen_length)
29
30
```

FIGURE B.3 – Script de génération de textes (*generate.py*)

INDEX

Epoch, 41
Evaluation, 19, 53
 Critères d'évaluation, 54
 Evaluation automatique, 21
 Evaluation humaine, 21, 53
 Formulaire d'évaluation, 54
 Métrique d'évaluation, 21

Modèles, 34
 Catégorisation des erreurs, 46
Métrique d'évaluation
 BLEU, 21
 ROUGE, 21

Phénomène d'hallucination, 19

Réseau de neurones, 41
 Entraînement, 42

Scribt : un outil Labsense, 24
 interface d'administration de projet, 24, 38
 le backoffice, 24, 36
synonymie, 36
Systèmes de GAT, 16
 BabyTalk, 17
 BRUTUS, 16
 ELIZA, 16
 JAPE, 16
 SHEFF2, 20
 STOP, 21
 TGEN, 20

textgenrnn, 41