Aix*Marseille université
Socialement engagée

NNT/NL : 2024AIXM0162/014ED184

THÈSE DE DOCTORAT

Soutenue à Aix-Marseille Université
le 05 juillet 2024 par

Adueni Adjoba Eunice AKANI

Factualité des résumés issus de modèles génératifs dans un contexte sémantique contrôlé

Discipline
Informatique

École doctorale
ED 184 MATHEMATIQUES ET INFORMATIQUE

Laboratoire/Partenaires de recherche
Laboratoire d'Informatique & des Systèmes (LIS)
ENEDIS

Composition du jury

Benjamin LECOUTEUX LIG, Univ. Grenoble Alpes	Rapporteur
Florian BOUDIN LS2N, Univ. de Nantes	Rapporteur
Géraldine DAMNATI Orange Innovation Lannion	Examinatrice
Richard DUFOUR LIA, Avignon Univ.	Examinateur
Cécile CAPPONI LIS, Aix-Marseille Univ.	Présidente du jury
Frédéric BECHET LIS, Aix-Marseille Univ.	Directeur de thèse
Benoit FAVRE LIS, Aix-Marseille Univ.	Co-Directeur de thèse

ENEDIS LIS TALEP

Affidavit

I, undersigned, Adueni Adjoba Eunice AKANI, hereby declare that the work presented in this manuscript is my own work, carried out under the scientific direction of Frédéric BECHET and Benoit FAVRE, in accordance with the principles of honesty, integrity and responsibility inherent to the research mission. The research work and the writing of this manuscript have been carried out in compliance with both the french national charter for Research Integrity and the Aix-Marseille University charter on the fight against plagiarism.

This work has not been submitted previously either in this country or in other country in the same or in a similar version to any other examination body.

Marseille 2024, 30 april



Cette œuvre est mise à disposition selon les termes de la Licence Creative Commons Attribution - Pas d'Utilisation Commerciale - Pas de Modification 4.0 International.

Liste de publications et participation aux conférences

Liste des publications réalisées dans le cadre du projet de thèse:

1. **Eunice Akani**, Benoit Favre, and Frederic Bechet. 2022. Abstraction ou hallucination ? État des lieux et évaluation du risque pour les modèles de génération de résumés automatiques de type séquence-à-séquence. In Actes de la 29e Conférence sur le Traitement Automatique des Langues Naturelles. Volume 1 : conférence principale, pages 2–11, Avignon, France. ATALA.
2. **E. Akani**, R. Gemignani and R. Abrougui, "EneBERT: a state-of-the-art language model trained on a corpus of texts generated from the set of DSO activities," 27th International Conference on Electricity Distribution (CIRED 2023), Rome, Italy, 2023, pp. 2903-2907, doi: 10.1049/icp.2023.0986.
3. R. Gemignani, **E. Akani**, J. P. Delrieux and A. Sayouti Souleymane, "Hape: optimizing customer relation by automatic task distribution using constrained optimization and natural language processing," 27th International Conference on Electricity Distribution (CIRED 2023), Rome, Italy, 2023, pp. 1764-1768, doi: 10.1049/icp.2023.1011.
4. **Eunice Akani**. 2023. Étude de la fidélité des entités dans les résumés par abstraction. In Actes de CORIA-TALN 2023. Actes des 16e Rencontres Jeunes Chercheurs en RI (RJCRI) et 25e Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL), pages 21–36, Paris, France. ATALA.
5. **Eunice Akani**, Benoit Favre, Frederic Bechet, and Romain Gemignani. 2023. Reducing named entity hallucination risk to ensure faithful summary generation. In Proceedings of the 16th International Natural Language Generation Conference, pages 437–442, Prague, Czechia. Association for Computational Linguistics.
6. **Eunice Akani**, Benoit Favre, Frederic Bechet, and Romain Gemignani. 2024. Approche multitâche pour l'amélioration de la fiabilité des systèmes de résumé automatique de conversation. JEP-TALN 2024.
7. **Eunice Akani**, Frederic Bechet, Benoit Favre, and Romain Gemignani. 2024. Unified Framework for Spoken Language Understanding and Summarization in Task-Based Human Dialog processing. INTERSPEECH 2024.

Participation aux conférences et écoles d'été au cours de la période de thèse:

1. *Advanced Language Processing Winter School,* ALPS 2021
2. *Conférence sur le Traitement Automatique des Langues Naturelles,* TALN 2022
3. *Language Resources and Evaluation Conference,* LREC 2022
4. *Journées Jointes des Groupements de Recherche " Linguistique Informatique, Formelle et de Terrain " et " Traitement Automatique des Langues ",* LIFT - TAL 2022
5. *Conférence sur le Traitement Automatique des Langues Naturelles,* CORIA - TALN 2023
6. *International Conference & Exhibition on Electricity Distribution,* CIRED 2023
7. *International Natural Language Generation Conference,* INLG - SIGDIAL 2023
8. *Conférence sur le Traitement Automatique des Langues Naturelles,* JEP - TALN 2024

Résumé

Le résumé automatique de document consiste à générer un résumé bref et cohérent de son contenu. Alors que les progrès récents des modèles de langue ont considérablement amélioré la capacité des systèmes à générer des résumés cohérents linguistiquement, maintenir une fidélité totale au contenu original demeure un défi pour les modèles génératifs. En effet, ces modèles vont paraphraser le texte à résumer, éventuellement en faisant d'autres choix lexicaux, favorisant ainsi la génération d'information non présente dans le document source. Cependant, avoir un résumé fidèle à ce document source est crucial, notamment dans un contexte industriel où des erreurs sur des valeurs ou des noms d'entités peuvent avoir des conséquences graves. C'est sur cette problématique que porte cette thèse. Elle l'aborde en deux volets principaux.

Tout d'abord, dans le cadre du résumé automatique de texte par abstraction, elle examine le phénomène des " hallucinations " en analysant les erreurs générées par les modèles de langue et en proposant des techniques pour réduire ces incohérences. Une typologie des erreurs pour les résumés automatiques, et une caractérisation du phénomène de l'abstraction pour les résumés de référence ont été proposées pour mieux comprendre l'ampleur de ces phénomènes. L'analyse des erreurs a montré que les systèmes sont sujets à générer des informations non-présentes, en particulier en ce qui concerne les entités nommées. Ainsi, basé sur l'hypothèse que les entités hors document sont plus susceptibles d'être des hallucinations, un nouveau critère de sélection de résumés est proposé. Ce critère est basé sur le "risque" de générer des entités qui n'appartiennent pas au document source. L'idée est de sélectionner, parmi un ensemble de résumé généré pour un même texte, le résumé qui a le moins d'entités potentiellement hallucinées.

Ensuite, cette thèse aborde le défi de faire un résumé de transcriptions de conversations entre plusieurs interlocuteurs. Cette tâche nécessite de considérer l'interaction entre les locuteurs pour préserver les informations les plus pertinentes du dialogue. Dans un contexte de communication humain-humain avec but, il est possible d'intégrer des informations relatives à la tâche pour faciliter le processus de résumé automatique de dialogue, ce qui peut aider les systèmes à produire des résumés plus fidèles sémantiquement. L'étude examine également les approches multitâches reliant la tâche de résumé à des tâches de compréhension du langage, telles que la détection de motifs d'appels d'un service client. Les informations liées à la tâche servent également de base pour proposer de nouvelles méthodes de sélection de résumés, fondées sur une analyse sémantique approfondie de la tâche liée aux dialogues, ainsi que des métriques d'évaluation basées sur cette même analyse.

Enfin, cette thèse s'est effectuée dans un contexte industriel à Enedis, entreprise

de distribution d'électricité qui réalise des millions d'interventions sur le réseau électrique chaque année. La problématique de la fiabilité des systèmes de résumé automatique est importante pour Enedis afin de synthétiser les interventions représentées par plusieurs documents pour identifier plus facilement les causes des interventions vaines et les anticiper. Même si toutes les expériences reportées dans ce document ont été menées sur des corpus publics, c'est dans ce cadre que cette thèse a étudié les méthodes liées au résumé de document (dialogue ou texte) afin de fiabiliser la génération d'un résumé cohérent et fidèle au document source.

Mots clés : Résumé de texte, Résumé dialogue orienté tâche, Hallucination, Fidélité des résumés, Entités nommées

Abstract

Automatic document summarization involves generating a concise and coherent summary of its content. While recent progress in language models have significantly improved the ability of systems to generate linguistically coherent summaries, maintaining full fidelity to the original content remains a challenge for generative models. Indeed, these models will paraphrase the text to be summarized, potentially making different lexical choices, thus increasing the risk of generating information not present in the source document. However, having a summary faithful to this source document is crucial, especially in an industrial context where errors in values or entity names can have serious consequences. This is the focus of this thesis, which addresses it in two main aspects.

Firstly, within the framework of automatic text summarization by abstraction, it examines the phenomenon of "hallucinations" by analyzing errors generated by language models and proposing techniques to reduce these inconsistencies. A typology of errors for automatic summaries, and a characterization of the abstraction phenomenon for reference summaries have been proposed to better understand the extent of these phenomena. Error analysis has shown that systems are prone to generating non-present information, especially concerning named entities. Thus, a new summary selection criterion is proposed based on the hypothesis that out-of-document entities are more likely to be hallucinations. This criterion is based on the "risk" of generating entities that not belonging to the source document. The idea is to select, among a set of summaries generated for the same text, the summary with the fewest “risky” entities.

Secondly, this thesis addresses the challenge of summarizing transcripts of conversations between multiple speakers. This task requires considering the interaction between speakers to preserve the most relevant information from the dialogue. In a context of human-human communication with a goal, it is possible to integrate task-related information to facilitate the automatic dialogue summarization process, which can assist systems in producing semantically more faithful summaries. The study also examines multitask approaches that link the summarization task to language comprehension tasks, such as detecting call types of conversations from a customer service. Task-related information also serves as the basis for proposing new summary selection methods based on a thorough semantic analysis of the task related to the dialogues, as well as evaluation metrics based on this same analysis.

Finally, this thesis was conducted in an industrial context at Enedis, a Distribution System Operator (DSO) company that carries out millions of interventions on the electricity network each year. The issue of the reliability of automatic summarization systems is important for Enedis to synthesize interventions represented by multiple

documents in order to identify the causes of unsuccessful interventions and anticipate them more easily. Although all experiments reported in this document were conducted on public corpora, it is in this context that this thesis studied methods related to document summarization (dialogue or text) in order to improve the generation of a coherent and faithful summary of the source document.

Keywords: Text summarization, Task-oriented dialog summarization, Hallucination, Faithfulness in summarization, Named entities

Résumé étendu

Le résumé automatique de texte consiste à faire une synthèse d'un document en gardant les informations pertinentes. Il existe deux types de résumé. Le résumé par extraction qui consiste à extraire les informations les plus pertinentes d'un document et à les agencer pour en faire un résumé ; et le résumé par abstraction qui consiste à faire la synthèse d'un document en utilisant des paraphrases et des nouveaux mots. Résumer un texte est une tâche complexe qui doit prendre en compte plusieurs critères. Il s'agit du but du résumé (le résumé peut être indicatif, informatif ou critique selon l'objectif de l'auteur), de la forme du résumé (extractive où les infos les plus critiques sont prises ou par abstraction avec une reformulation complète), de la dimension du résumé (le résumé peut varier selon qu'il s'agit d'un seul document ou de plusieurs documents) et enfin du contexte du résumé (il peut être générique ou personnalisé selon les besoins du lecteur). Aujourd'hui, à l'ère du numérique, nous devons traiter un grand nombre d'informations. Ainsi, avoir un résumé de toutes ces informations permet de les traiter plus facilement pour éviter la surcharge de données. De plus en industrie, le résumé facilite la compréhension rapide d'un document écrit, d'un événement ou d'une tâche accomplie. Il est donc précieux pour gagner du temps. L'automatisation à l'aide de système automatique est donc nécessaire. Les systèmes automatiques sont conçus pour accomplir des tâches sans intervention humaine. Pour le résumé de texte, ces systèmes sont entraînés sur des paires document-résumé afin de produire des résumés concis. Cependant, l'utilisation de données de résumé abstrait (dans le cadre du résumé abstractif lors de l'entraînement) peut engendrer une incohérence entre le résumé généré et le contenu du document source. Dans la littérature, ces incohérences sont appelées des *hallucinations*. Les *hallucinations* dans les résumés de texte sont dues à la nature des systèmes utilisés. Ces systèmes fonctionnent à partir de modèles génératifs capables de produire un texte similaire à celui écrit par un humain. Ils apprennent les structures et les schémas linguistiques à partir d'un large ensemble de données textuelles, puis utilisent ces connaissances pour générer de nouveaux textes pour diverses applications comme la traduction, la création de contenu, les chatbots et bien sûr, le résumé de texte. Ces inconsistances peuvent créer de gros problèmes notamment dans le domaine de la santé où des erreurs peuvent être fatales. De plus, le résumé ne remplit plus sa fonction s'il contient des inconsistances. C'est sur cette thématique que porte cette thèse. La principale question que nous tentons de résoudre est :

Comment s'assurer que le résumé généré reste fidèle au document source ?

Pour répondre à cette question, nous nous sommes concentrés sur la modification du processus de génération des résumés pour éviter les inconsistances. Dans un pre-

mier temps, nous avons proposé une typologie des erreurs dans le résumé afin de les comprendre et une typologie d'abstraction pour les résumés de référence pour comprendre le comportement des modèles. Cette étude nous a permis de nous focaliser sur les entités nommées qui sont une manière simple de représenter l'information dans un texte. Plusieurs hypothèses ont été alors explorées :

— Est-ce que les entités nommées du résumé en dehors du document sont en majorité des hallucinations ? Nous avons étudié les entités nommées de plusieurs systèmes.

— Est-ce que réduire le nombre d'entités en dehors du document permet d'avoir un résumé fidèle au document source ? Nous avons décidé d'explorer deux méthodes pour contraindre les systèmes à générer des entités nommées provenant uniquement du document.

Puis nous nous sommes concentrés sur le résumé de dialogue orienté tâche qui, contrairement au texte, nécessite une compréhension plus approfondie du dialogue parce qu'il implique plusieurs locuteurs qui interagissent de manière asynchronisée pour accomplir une tâche. Les hypothèses de cette partie sont :

— Intégrer des informations sémantiques dans un résumé de dialogue orienté tâche produit-il un résumé cohérent avec la conversation ?

— L'utilisation d'un système automatique pour générer de nouvelles données affecte-t-elle la qualité des résumés par rapport à la conversation ?

— Les systèmes de transcription automatique peuvent-ils réduire les coûts tout en assurant la cohérence des résumés ?

Outre le **Chapitre 1** qui examine l'état de l'art et introduit la problématique abordée dans cette thèse, trois autres chapitres sont inclus pour traiter chacune des questions mentionnées précédemment.

Chapitre 2 : Typologie des erreurs dans le résumé automatique de texte

Ce chapitre vise à répondre à deux questions. La première est comment les erreurs dans le résumé automatique sont-elles catégorisées ? Pour ce faire, nous proposons non seulement une typologie des erreurs dans les résumés générés automatiquement mais également une catégorisation du phénomène d'abstraction dans les résumés de référence. En effet, les travaux antérieurs considéraient le phénomène d'abstraction dans les résumés de référence comme des erreurs ou des hallucinations. En ce qui nous concerne, nous distinguons les hallucinations, un type d'erreur dans les résumés générés de l'abstraction qui est une représentation simplifiée de concept ou d'ensemble d'informations linguistiques. Nous proposons 6 types d'erreurs et 3 différents types d'abstraction. Puis, nous avons conduit une évaluation humaine de résumés générés par 4 différents systèmes de l'état de l'art (à l'époque des expériences) basés sur l'architecture des RNN et sur celle des Transformers afin d'y analyser les erreurs.

Ces systèmes ont été entraîné sur le jeu de données Orange-Sum. L'analyse des erreurs de chaque système a révélé que les modèles pré-entraînés génèrent des informations qui ne sont pas dans le document source. En ce qui concerne les résumés de référence, notre analyse a montré que la plupart des abstractions dans ceux-ci ne sont pas inférables. Ce qui motive les modèles à générer des informations en dehors du document source.

La seconde question que nous avons traitée dans ce chapitre concerne la représentation des informations dans le résumé. En effet, pour une étude plus fine des hallucinations, il nous fallait choisir un élément linguistique qui représente de manière explicite une séquence de mots. Nous nous sommes concentrés sur les entités nommées car ces dernières sont des informations qu'on peut extraire facilement du texte grâce à l'avancée des travaux sur la détection des entités nommées. Nous avons donc analysé les sorties des différents systèmes en faisant un focus sur les entités nommées. Notre analyse a montré que lorsqu'une entité ne provient pas du document source, la plupart du temps elle est hallucinée. Ces résultats nous permettent ainsi de poser l'hypothèse selon laquelle avoir des entités nommées provenant uniquement du document source permettrait d'éviter les hallucinations. C'est cette hypothèse que le chapitre suivant aborde.

Chapitre 3 : Utilisation du risque d'hallucination pour réduire les hallucinations sur les entités nommées

De notre analyse au chapitre 2, nous avons decidé de nous concentrer sur l'hypothèse selon laquelle il est mieux d'avoir des entités nommées qui proviennent du document source pour éviter les hallucinations. Pour ce faire, nous avons, dans un premier temps proposé une métrique basée sur le risque d'avoir des entités hallucinées, le NEHR dont l'avantage est d'être une métrique qui ne nécessite pas de supervision ou une profonde compréhension du texte. Cette métrique calcule le pourcentage d'entités nommées en dehors du document source. Une étude empirique du critère a été effectuée afin d'évaluer la pertinence de cette métrique. L'étude consistait à évaluer manuellement les entités nommées de résumés ayant le score ROUGE maximal parmi un ensemble de résumés générés en utilisant plusieurs techniques de décodage. Cette analyse a montré qu'environ 20% des entités nommées étaient en dehors du document et parmi elles uniquement 29% étaient inférables. En ce qui concerne les entités nommées provenant du document source, notre analyse a montré que lorsqu'une entité nommée est dans le document source, en grande majorité, 90% du temps elle est utilisée dans le bon contexte. De plus, 40% des entités nommées provenant du document étaient des variations d'entités nommées du document source. Ce qui nous a conduits à proposer plusieurs heuristiques afin de détecter automatiquement les variations, et ainsi garantir une fiabilité de notre métrique sur les entités nommées. A la vue des analyses, nous avons décidé de forcer les systèmes automatiques à ne

prédire que des entités nommées provenant du document source. Deux méthodes basées sur le NEHR ont été proposées. La première consistait à sélectionner parmi un ensemble de résumés générés à l'aide de techniques de décodage, pour un même exemple, celui qui minimise le NEHR. L'analyse des résultats a montré que cette piste était prometteuse. D'ailleurs, une évaluation humaine des entités nommées a montré que les résumés sélectionnés selon ce critère étaient utilisés dans un bon contexte par rapport au modèle de référence. Ensuite, la seconde consistait à pénaliser le modèle pendant la phase d'apprentissage lorsqu'il prédisait des entités nommées en dehors du document source ou de la référence. Ainsi, trois différentes fonctions de coût pour l'entraînement ont été créées. Cette piste n'a pas donné les résultats attendus et sera explorée dans nos futurs travaux. Nous avons montré que le NEHR permet d'éviter les hallucinations au niveau des entités nommées. Cette étude a été réalisée sur le résumé de texte de manière générale. Pour la suite, nous nous sommes concentrés sur un cas particulier de résumé automatique : il s'agit du résumé automatique de dialogue orienté tâche.

Chapitre 4 : Amélioration de la fidélité dans le résumé de dialogue

Dans ce chapitre, nous abordons le défi de faire un résumé de transcriptions de conversations entre plusieurs interlocuteurs. Cette tâche nécessite de considérer l'interaction entre les locuteurs pour préserver les informations les plus pertinentes du dialogue. Un des challenges du résumé automatique de dialogue provient de la différence entre le style du dialogue et le résumé. Des erreurs de transcriptions, les disfluences et le processus de génération en lui-même peuvent conduire à des inconsistances du résumés par rapport au dialogue. Un autre challenge est le manque de données annotées pour l'utilisation de méthode supervisée. En effet, en utilisant une méthode pipeline pour le résumé de dialogue, il nous faut avoir non seulement l'audio brut et le résumé mais également la transcription de l'audio. Ce chapitre tente de répondre à la question : comment avoir un résumé fidèle au dialogue et particulièrement pour le dialogue orienté tâche ? Pour répondre à cette question nous avons exploré plusieurs hypothèses en utilisant le jeu de données DECODA qui est le résultat de la conversation entre un agent et un usager du réseau de la RATP. D'abord, nous avons étudié comment l'intégration des informations sémantiques peuvent aider à avoir un résumé sémantique fidèle au document source au niveau de l'information. Pour ce faire, nous avons proposé d'intégrer des informations sémantiques qui ont été proposées pour la tâche de compréhension du langage parlé dans le résumé de texte. Il s'agit du domaine (Transport public), l'intention (le motif de l'appel) et la paire entité-valeur (les entités nommées spécifiques à DECODA). Pour l'intégration, nous avons proposé deux manières d'intégrer les différentes informations :

— Pendant l'entrainement du modèle : à l'entrée du modèle en pipeline et la sortie du modèle en multitâche (le système doit d'abord générer l'information

sémantique avant le résumé). Nous avons également proposé des métriques d'évaluation du résumé basées sur ces informations sémantiques.

— Pendant la phase de génération : en introduisant un critère de sélection de résumé le plus fidèle suivant les informations sémantiques énoncées plus haut (le motif d'appel et les entités nommées).

Nous avons aussi proposé une métrique sur les motifs d'appel basée sur l'hypothèse qu'un résumé généré doit avoir le même motif d'appel que le résumé de référence et trois métriques sur les entités nommées qui évaluent la capacité des systèmes à générer des entités par rapport au résumé de référence. Les résultats ont montré que la sélection de résumés par les différents critères permettait d'améliorer nos scores sémantiques.

Ensuite, nous avons abordé la question du manque de données en proposant une nouvelle version du corpus DECODA qui a été faite en utilisant un LLM pour générer les résumés manquants et normaliser les données écrites sous une forme non littéraire. De plus, nous nous sommes placés dans un cadre où nous n'avons pas accès au transcription manuelle et avons évalué l'impact de l'utilisation des systèmes de transcription automatique comme WhisperX. L'analyse de l'impact de l'augmentation de données nous permet d'avoir une légère augmentation des scores du résumé automatique. Mais, les résumés obtenus sont plus informatifs et peuvent être utilisés pour l'approfondissement du résumé automatique orienté tâche. En ce qui concerne les transcriptions automatiques, la qualité des résumés obtenus est inversement corrélée au taux d'erreurs mots (WER) des transcriptions. En effet, lorsque le WER score diminue, les scores sur les résumés augmentent et les scores sémantiques également.

Enfin nous avons étudié l'impact de l'intégration des informations sémantiques sur la nouvelle version de DECODA. L'analyse des résultats a montré que les scores sémantiques augmentent moins lorsque l'on applique la sélection de résumé. Cela peut être dû au fait que les résumés augmentés sont de meilleures qualités et donc permettent une conservation de la sémantique du dialogue. De plus, nous avons testé la sélection de résumé sur un système avec un WER score élevé et avons montré que, dans ce contexte, la sélection de résumés permet d'améliorer la sémantique du dialogue en ce que les scores sémantiques sont élevés. Des évaluations humaines ainsi que des calculs de significativité restent à faire.

Conclusion

Tout au long de cette thèse nous avons tenté de répondre à la question : Comment garantir la fidélité des résumés générés par rapport au document source? Cette question est d'un enjeu majeur surtout de nos jours, à l'ère du numérique où nous avons une quantité importante d'informations à traiter. En effet, avoir des résumés de ces informations est cruciale afin de ne pas être déborder. De plus, cette thématique a également son importance en industrie mais aussi dans le secteur de la santé où un résumé non fidèle au document source peut avoir des conséquences mortelles. Nous

avons abordé cette thématique suivant deux volets. Le premier concerne le résumé de texte en général et le second s'accentue sur le résumé de dialogue orienté tâche.

Pour ce qui concerne le résumé de texte, l'analyse des erreurs dans les résumés de plusieurs systèmes, celle des abstractions dans le résumé de référence et particulièrement l'analyse des entités nommées a permis de montrer que les systèmes hallucinent sur les entités nommées lorsqu'ils essaient de faire des abstractions. Cette étude nous a conduit à forcer les systèmes à générer des entités nommées provenant uniquement du document source. Pour y parvenir, nous avons dans un premier temps introduit une métrique basée sur le pourcentage d'entités en dehors du document source. Puis nous avons utilisé cette métrique pour sélectionner les résumés avec le moins d'entités en dehors du document parmi des résumés générés par l'usage de plusieurs méthodes de décodage. Cette métrique a également été utilisée pour modifier la fonction de coût d'entrainement des systèmes. Les résultats de l'évaluation humaine montrent que l'utilisation du critère de sélection est une solution prometteuse.

Pour ce qui est du résumé de dialogue, nous avons intégré des informations sémantiques dans la tâche de résumé automatique, pour améliorer ainsi sa fidélité. Deux approches ont été testées : une intégration pendant la phase d'entraînement et celle de génération à travers l'utilisation de critères de sélection basés sur les informations sémantiques. Les résultats obtenus sont prometteurs et montrent le bon fonctionnement de la sélection de résumé. Nous avons enrichi l'ensemble de données DECODA et exploré l'utilisation de la transcription automatique pour générer les résumés. Nos résultats montrent que l'intégration d'informations sémantiques améliore la qualité des résumés, et que la transcription automatique peut être une alternative viable à la transcription manuelle. Cependant, des évaluations humaines sont nécessaires pour confirmer ces conclusions.

Perspectives

À la suite de notre étude, les perspectives suivantes sont envisagées :

— *Amélioration du processus d'entraînement des modèles génératifs :* Nous proposons d'optimiser la fonction de coût utilisée lors de l'entrainement du modèle par l'usage d'un système de renforcement basé sur les critères NEHR ou d'autres approches telles que celles basées sur du Question Answering (QA). De plus, nous pouvons explorer l'intégration de technique d'apprentissage par renforcement à partir de rétroaction humaine (RLHF) pour améliorer l'entraînement en utilisant des retours humains sur la fidélité du résumé généré.

— *Modification du processus de décodage et évaluation des hallucinations :* Nous souhaitons explorer la méthode proposée par [CCK22], en changeant les scores du discriminateur pour guider la génération de texte et réduire les hallucinations.

— *Exploration d'autres types d'informations pour améliorer la fidélité des résumés :* Nous envisageons d'explorer l'intégration des cadres sémantiques (FrameNet), des unités sémantiques (SCU), et de la structure du dialogue pour améliorer la fidélité du résumé.

— *Exploration des méthodes end-to-end pour le résumé de dialogue orienté tâche :* Nous voulons étudier l'impact de l'utilisation d'une approche de bout en bout pour la synthèse des dialogues, en utilisant des méthodes telles que Whisper pour résumer de l'audio brut.

— *Étude du compromis qualité-calcul avec différentes tailles de LLM :* Nous prévoyons de répéter nos expériences avec les derniers LLM, notamment LLama et Mistral, pour améliorer la qualité des résumés. L'utilisation de LLM open source pourrait être une alternative pertinente pour normaliser et augmenter les ensembles de données, en particulier lorsque des données confidentielles sont impliquées.

Remerciements

Il m'est impossible de citer individuellement toutes les personnes dont le soutien a été crucial à l'aboutissement de cette thèse, tant elles ont été nombreuses et essentielles.

Je souhaite tout d'abord exprimer ma gratitude à Jireh, mon TouT, sans qui cette aventure n'aurait jamais été possible. Il a pourvu à cette thèse et à mes besoins. Il m'a accompagné tout au long de ces années, me prodiguant une aide précieuse et indéfectible, même dans les moments les plus difficiles.

Je tiens ensuite à remercier profondément mes directeurs de thèse, Frédéric Bechet et Benoît Favre, pour leur accompagnement bienveillant tout au long de ce parcours académique. Leur expertise, leurs conseils avisés, ainsi que leur disponibilité ont été des éléments indispensables à la réalisation de ce travail. Vos remarques rigoureuses et vos encouragements m'ont constamment poussée à donner le meilleur de moi-même.

Je suis également très reconnaissante aux membres du jury d'avoir pris part à l'évaluation de cette thèse. Je remercie tout particulièrement les rapporteurs, Pr Benjamin Lecouteux et Dr Florian Boudin, pour leurs analyses minutieuses et leurs commentaires enrichissants. Mes remerciements vont également aux examinateurs, Pr Richard Dufour et Dr Géraldine Damnati, pour leur lecture attentive et leurs questions stimulantes lors de la soutenance. Enfin, je remercie chaleureusement Pr Cécile Capponi d'avoir présidé ce jury avec bienveillance et pour l'intérêt qu'elle a porté à cette recherche.

Un grand merci à Yannick Estève et Carlos Ramisch, membres du comité de suivi de thèse, pour leur accompagnement tout au long de ces trois années. Yannick, tu as été un véritable pilier et tes encouragements m'ont permis de persévérer. Carlos, tes conseils avisés sur l'équilibre entre vie privée et recherche ont été précieux pour mener cette thèse à bien.

Cette thèse s'inscrivant dans un contexte industriel, je souhaite exprimer toute ma gratitude à Romain Gemignani de m'avoir offert cette opportunité au sein d'Enedis. Merci pour la confiance que tu m'as accordée. Je tiens également à remercier tous les collaborateurs d'Enedis avec qui j'ai eu la chance de travailler pendant ces années : Xavier, Dorothée, Aubin Porte, Maëva, Abdoulaye, Chérif, Hugo, Nathan, Abdelahdi, Jean-Luc, Ekaterina, et tant d'autres.

Je n'oublie pas non plus la CISAM et toutes les personnes qui y travaillent, dont la

présence a enrichi cette expérience humaine.

Mes remerciements vont également au LIS et à l'équipe TALEP de m'avoir accueillie au sein du laboratoire. Bien que n'ayant pas à temps plein, j'ai eu beaucoup de plaisir à partager ces années avec vous. Un remerciement spécial à Emmanuelle, François et Élie pour les moments de détente partagés autour de nos parties de The crew et Skyjo.

Je remercie de tout cœur ceux qui ont contribué aux différentes tâches d'annotation : Élie, Jean-Luc, Loredana, Nicaise, Cédric et tant d'autres. Vos contributions se reflètent dans chaque page de cette thèse. Je tiens également à remercier chaleureusement ceux qui ont pris le temps de relire et de corriger ce manuscrit : Yann, Ismaël, Cédric, Loredana, Nicaise, Élodie, et bien d'autres.

Je souhaite adresser des remerciements tout particuliers à ma famille. À mon père, ma mère, et ma sœur Élodie, votre soutien inébranlable, malgré la distance, m'a porté. Vous avez cru en moi et en ma capacité à mener ce projet à terme. Mon frère Nicaise, ta présence m'a aidé à garder le cap et à rester concentrée tout au long de cette aventure.

À Loredana (Eema comme j'aime t'appeler), je te suis infiniment reconnaissante. Tu as été au cœur de cette aventure, dans la joie comme dans les moments difficiles. Ton soutien, tes conseils, et ta présence constante ont été une véritable source de force pour moi. Merci infiniment.

À Christophe, merci de m'avoir orientée vers la Data dès ma licence. C'est grâce à tes conseils que je me suis tournée vers ce domaine. Et aujoud'hui, je suis docteure.

Je n'oublie pas non plus mon ecclésia, ma deuxième famille (les Pastor, les Ensminger, les Sorrenti, les Guillaume, les Amic, les Bouyou, Pierre-Laurent, Cédric, Océanne, Alexis, le groupe de jeune, etc.), qui m'a portée tout au long de cette période et qui s'est mobilisée pour m'accompagner lors de la soutenance.

Pour finir, je souhaite remercier chaleureusement tous ceux qui, d'une manière ou d'une autre, ont croisé ma route durant ces années et m'ont apporté leur aide, leur soutien, leur amitié, leur gentillesse et leur joie : Chérif, Lauryn, Jemima, Romaric, Sosthène, et tant d'autres que je n'ai peut-être pas nommés ici, mais qui restent gravés dans mon cœur.

Merci pour tout! ♡

Table des matières

Affidavit 2

Liste de publications et participation aux conférences 3

Résumé 5

Abstract 7

Résumé étendu 9

Remerciements 16

Table des matières 18

Table des figures 21

Liste des tableaux 22

Liste des acronymes 25

Introduction 28

1 Automatic summarization: State of the art and problematic of the study 34

1.1 Introduction . . . 34
1.2 Automatic text summarization . . . 36
1.2.1 Text summarization datasets . . . 36
1.2.2 Text summarization Approaches . . . 40
1.3 Dialog summarization . . . 51
1.3.1 Definitions . . . 51
1.3.2 Dialog dataset for summarization . . . 52
1.3.3 Approaches for dialog summarization . . . 54
1.4 Automatic Evaluation measures . . . 57
1.5 Hallucination in automatic summarization . . . 59
1.5.1 Definition . . . 60
1.5.2 Hallucination metrics . . . 61
1.5.3 Methods to relieve hallucination in text summarization . . . 66
1.5.4 Hallucination in dialog summarization . . . 68
1.6 Conclusion . . . 69

2 Typology of errors in text automatic summarization 70
2.1 Introduction . . . 70
2.2 Typology Definition . . . 71
2.2.1 Error typology . . . 71
2.2.2 Abstraction typology . . . 73
2.3 Evaluating summaries with the proposed typology . . . 74
2.3.1 Experimental setup . . . 74
2.3.2 Evaluation of generated summaries . . . 76
2.3.3 Study of named entities in generated summaries . . . 79
2.4 Discussion . . . 82
2.5 Conclusion . . . 83

3 Improving summary faithfulness by minimization entity hallucination risk 85
3.1 Introduction . . . 86
3.2 Named entity hallucination risk . . . 86
3.2.1 Definition . . . 87
3.2.2 Assessing the relevance of the NEHR criterion . . . 87
3.2.3 Empirical study on the NEHR criterion . . . 88
3.2.4 Entities variation detection . . . 93
3.3 Using NEHR as a summary selection criterion . . . 97
3.3.1 Selection of summaries . . . 97
3.3.2 Evaluate the impact of summary selection based on NEHR . . . 97
3.3.3 Discussion . . . 99
3.4 Using loss function based on NEHR to ensure entities faithfulness . . . 100
3.4.1 Machine learning: training steps . . . 100
3.4.2 Proposed method . . . 101
3.4.3 Impact of Loss function based on NEHR score on the summary faithfulness . . . 106
3.4.4 Discussion . . . 109
3.5 Conclusion and Discussion . . . 110

4 Improving faithfulness in dialogue summarization 112
4.1 Introduction . . . 113
4.2 Dialog summarization approaches . . . 114
4.3 DECODA Corpus . . . 116
4.3.1 Description . . . 116
4.3.2 Details on DECODA and statistics . . . 116
4.4 Integrating Spoken Language Understanding and Summarization . . . 120
4.4.1 Integrating semantic information to train summarization model 120
4.4.2 Task semantic information as a summary selection process . . . 122
4.4.3 Evaluation metric based on SLU . . . 124
4.4.4 Experiments and results . . . 126
4.4.5 Discussion . . . 132

4.5 Data normalisation and data augmentation for spoken conversation summarization . . . 133
4.5.1 Motivation . . . 133
4.5.2 Data normalisation and Data augmentation methods . . . 134
4.5.3 Experiments and results . . . 138
4.5.4 Discussion . . . 141
4.6 SLU and data augmentation for dialog summarization . . . 142
4.6.1 Integrate call type and named entities . . . 142
4.6.2 Experiments and results . . . 142
4.6.3 Discussion . . . 146
4.7 Conclusion . . . 147

Conclusion **149**

Bibliography **153**

Table des figures

1.1 Example : Extractive vs Abstractive summarization 36
1.2 BART architecture [Lew+20] 49
1.3 PEGASUS architecture [Zha+20a] 50
1.4 Example of dialog 51
1.5 Example of extrinsic and intrinsic hallucination. [May+20] 60
1.6 Framework of QA and NLI based metrics 62

2.1 Example of errors type in summary 72
2.2 The different types of error in each system. 78
2.3 Example of abstraction type in Orange-Sum reference summary 79
2.4 Entities annotation example 80

3.1 Annotation procedure 92
3.2 Example of BERTScore for variation detection 95
3.3 Step to train a model 101
3.4 An example of XSum with named entity tag. 107

4.1 Pipeline method vs End-to-end method 115
4.2 DECODA call type distribution 117
4.3 Integration of semantic information about the task during summary generation process 121
4.4 Step to evaluate generated summary in term of CT-Acc 125
4.5 Prediction of call type on the training dataset. 128
4.6 Manual evaluation of the summaries generated by the baseline model, Multitask$_C$ and Pipeline$_C$. Percentage of times each system was ranked first in terms of faithfulness and informativeness. 131
4.7 Integration of semantic information about the task during summary generation process 143

Liste des tableaux

1.1 Statistics on abstractive text summarization dataset. ADW - Average Document Word, ADS - Average Document Sentence, ASW - Average Summary Word, ASS - Average Summary Sentence. The statistics of dataset with * was taken from the paper [BC20] Other statistics were generated by us. 39
1.2 Model - datasets 51
1.3 Dialog summarization datasets: Statistics, domains and languages . . . 54
1.4 Human evaluation factors 65
1.5 SUMMAC Benchmark [Lab+22]. Statistics of the different dataset : CoGenSum - Correctness of Generated Summaries [Fal+19], XSumFaith - eXtreme Summarization Faithfulness [May+20], Polytope [Hua+20], FactCC [Kry+20], SumEval [Fab+21], FRANK [PBT21] 66

2.1 Statistics about the dataset Orange-Sum. 75
2.2 Orange-Sum Title vs Orange-Sum Abstract example 75
2.3 ROUGE (ROUGE-1, ROUGE-2, ROUGE-L) et BERTScore F1 for the models. 77
2.4 Error statistic of the models 77
2.5 Percentage appearance of different types of abstraction in the 100 annotated reference abstracts. 79
2.6 Percentage of Out-Of-Document (OOD) entities and of potentially hallucinatory entities obtained using manual evaluation 81
2.7 Percentage of OOD entities and of potentially hallucinatory entities obtained using automatic approach 81
2.8 Examples of reference summary with out of document information. . . 82
2.9 List of contributions made in this chapter 84
2.10 List of papers based on this chapter 84

3.1 Sampling parameters 90
3.2 Description of OntoNotes Named Entities (NE)s. 91
3.3 Maximal and minimal values of ROUGE on the summary set produced by our sampling method on CNN/DM [Her+15] and XSum [NCL18]. The NEHR value for the min/max summaries is also reported. 91
3.4 Cohen kappa inter-annotator agreement [Coh60] between the 3 annotators. 93

3.5 Percentage of correctly used entities for each subset of in-document and out-of-document entities for CNN/DM dataset. Dist. is the percentage of entities belonging to each type for entities inside (*exact match, variation*) and outside (*inferable, hallucination*) the document. 93
3.6 Percentage of variations detected for each heuristic. 96
3.7 Accuracy, Recall and F1 score for detecting variation and OOD entities 96
3.8 Evaluation on CNNDM and XSum. BEST PROBA – the summary with the best score among all generated summaries given by the model. MIN NEHR – our proposed summary selection method after using variation heuristics. %HallSum – the percentage of summary with at least one entity out of the document. ↑ higher is better, ↓ lower is better. 98
3.9 Percentage of reference summaries with at least one entity outside the source document 99
3.10 Average percentage of well-used entities for each system annotated. In adjudicated results, the annotators had to agree on a label for each entity. 99
3.11 Summary of the different loss function proposed. 106
3.12 Data distribution. Comparison between the original Sum dataset and the selected part. 107
3.13 Evaluation of the generated summaries. R1, R2, RL are the different ROUGE score. Entities precision (ent-P) recall (ent-R) and f1 (ent-F1) between the candidate and the reference. The average number of entities (#ent-avg) in the summary. 109
3.14 List of contributions made in this chapter 110
3.15 List of papers based on this chapter 111

4.1 Description of DECODA corpus. #dialog: total number of dialogs; #turn: average number of turns; #conv.len and #sum.len average conversation and summary number of words respectively. 117
4.2 Difference between DECODA part 1 and part 3. Example 20091112_-RATP_SCD_0152 for part 1 and 20110704_RATP_SCD_0018 for part 3 from DECODA. 118
4.3 Description of DECODA NEs 119
4.4 Example of manual annotation of the DECODA corpus 119
4.5 DECODA dataset distribution. #examples is the number of examples for each partition. #conv. words and #syn. words refer to the average number of words in the conversation and the synopsis respectively. #conv. tokens and #syn. tokens refer to the average number of tokens for the conversation and the synopsys respectively. 126
4.6 Call type classification results using the conversations as input Conv-CT-classifier, the synopses as input Syn-CT-classifier and the multitask model Multitask$_C$. Acc. is the accuracy and W-F1 is the weighted average F1. 129
4.7 Difference between the different Pipeline$_C$ models. ∘: reference call type was taken. ×: predicted call type was taken 129

4.8 Evaluation results of text summarization system. R1, R2, RL correspond to different ROUGE-Score. CT-Acc is the accuracy of call type prediction. NE-P, NE-R and NE-F1 is the entity precision, recall and F1 respectively. 130
4.9 R1,R2,RL: ROUGE; CT-Acc: Call type prediction from synopses (ref=oracle synopses); NE-R and NE-P: Entities recall and precision of the summaries generated compared to the reference 132
4.10 Manual transcription vs Automatic transcription with Whisper large (Example 20091112_RATP_SCD_0152 of DECODA). 135
4.11 Data augmentation and data normalisation example (Example 20091112_-RATP_SCD_0152 of DECODA). 136
4.12 Decoda data distribution in the train (human, human + bullet point, human + normalisation, human + augmentation) and the test set . . . 139
4.13 RougeL and BERTScore on the test partition on the manual transcription of DECODA with Barthez fined tuned on the different datasets (Orange-Sum [KTV21], train-H, train-H+B, train-H+N, train-H+A). Performance of chatGPT3.5 is also presented. (size) refers to the number of DECODA example each system seen during training. 139
4.14 WER score on automatic transcription from different size of WhisperX: T-Tiny, B-base, S-small, M-medium, L-large. Rouge-L and BERT-scores on summaries generated by Barthez finetuned and evaluated on automatic transcriptions. CT-Acc, NE-P, NE-R and NE-F1 our semantic measure based on call type and named entities. 141
4.15 Accuracy and Weighted F1 Score of call types classifiers and generation systems 144
4.16 Summaries evaluation. R1, R2, R3 are ROUGE Score, BS is BERT-scores. CT-Acc, NE-P, NE-R, NE-F1 are our proposed semantic information. . . 144
4.17 Summaries evaluation of summary selection method. ChatGPT3.5 results are included 145
4.18 Summaries evaluation of summary selection method using automatic transcription provided by WhisperX base. 146
4.19 List of papers based on this chapter 148

Liste des acronymes

Agram
Agrammaticality. 72, 81

ASR
Automatic Speech Recognition. 114, 115, 134, 138, 140, 142, 151, 152

BERT
Bidirectional Encoder Representations from Transformers. 47–49, 59, 62, 63

C4
Colossal Clean Crawled Corpus. 49, 50

CNN
Convolutional Neural Network. 41, 42, 46, 62

DNN
Deep Neural Networks. 41

DUC
Document Understanding Conferences. 37

Ext-Hall
Extrinsic Hallucination. 60, 61

GRU
Gated Recurrent Unit. 43

HorsDoc
Outside the document. 72, 73, 77–79

IDF
Inverse Document Frequency. 43, 44

Int-Hall
Intrinsic Hallucination. 60, 61

KL
Kullback-Leibler. 122, 123, 127, 131–133

LLM
Large Language Model(s). 30, 114, 134–138, 144, 146, 147

LSTM
Long-Short Term Memory. 40, 42–44, 68

MCTS
Monte Carlo Tree Search. 110, 151

MLM
Masked Language Model. 48, 49

NE
Named Entities. 22, 23, 90, 91, 117, 119, 123, 127, 129, 142, 144

NEHR
Named Entity Hallucination Risk. 86–88, 90, 94, 96–100, 104, 105, 108–110, 113, 131–133

NER
Named Entity Recognition. 44, 80, 81, 86–88, 90, 107, 120, 128

NLI
Natural Lnaguage Generation. 61–64, 66, 67, 69

NLP
Natural Language Processing. 28, 30, 36, 37, 41, 46, 104, 134

NNLM
Neural Network Language Model. 41, 42

NonInf
Non-inferable. 72, 73, 78, 79

NP
Noun Phrase. 63

NSP
Next Sentence Prediction. 48

ODD
Open-Domain Dialog. 52

OOD
Out-Of-Document. 22, 23, 81, 83, 84, 86, 87, 92, 93, 95, 96, 98, 101, 102, 110, 123

OOV
Out-Of-Vocabulary. 44, 46

POS
Part Of Speech. 43, 44

QA
Question Answering. 61, 63, 64, 66, 110, 151

QG
Question Generation. 63

RefE
Reference Error. 72, 73

RL
Reinforcement Learning. 44–46, 66, 110, 151

RLHF
Reinforcement Learning from Human Feedback. 110, 151

RNN
Recurrent Neural Network. 40–44, 46, 74, 75

SCU
Summarization Content Unit. 59, 152

SemCtr
Semantic contradiction. 72, 78, 81

SLU
Spoken Language Understanding. 120, 150

T5
Text-To-Text Transfer Transformer. 49, 76

TF
Term Frequency. 43, 44

TGDGA
Topic-word Guided Dialogue Graph Attention. 56

TOD
Task-Oriented Dialog. 52

VAD
Voice Activity Detection. 134

WER
Word Error Rate. 140, 141, 146

Introduction

As per the definition on Wikipedia [1], a summary is a brief piece of writing that captures the essential points of a text in one or more paragraphs. Thus summarization involves condensing the most crucial information from a lengthy text into a shorter version. However, summarization is more complex than it seems. It can vary depending on several criteria, such as purpose, form, dimension, and context. For instance, summarizing a soccer match requires highlighting different aspects, such as scores, scorers, and penalties, which can vary from one person to another. These criteria have to be taken into account. The purpose of a summary can be indicative, informative, or critical; depending on what the author aims to achieve. The form of a summary can be extractive, where the most critical points are picked out and presented, or abstractive, where a complete reformulation of the text is created. The dimension of a summary can also vary, depending on whether it is a single-document summary or a multi-document summary. Furthermore, the context of the summary can be generic or personalized, depending on the user's needs. Creating a well-structured summary requires an essential understanding of the target audience and their requirements to provide a comprehensive and effective summary.

While summarizing text or information can be difficult, its usefulness is of vital importance. Today's society is experiencing groundbreaking advancements in the way information is disseminated, mainly due to digitization, recommendation systems, social networking, and other digital innovations. As a result, an overwhelming amount of data and information is being generated. To avoid information overload, it is necessary to summarize this data. Summarizing enables clear and effective presentation of information, promotes better understanding, conveys more explicit ideas, and saves valuable time. Moreover, in an industrial context, it facilitates swift comprehension of a written piece, an event, or a completed assignment. Summarization in this context can be an invaluable tool for saving time. Therefore, the automation of this task becomes all the more compelling. To this end, automatic text summarization, a Natural Language Processing (NLP) task, has been introduced.

Automatic text summarization involves rewriting a source document while preserving its most important information. This task enables rapid comprehension of a document by highlighting its key ideas. It can also be used to extract relevant information from a document to facilitate decision-making or research. The difficulty in setting up this task is nuanced. The first challenge is the nature of the document to be summarized. Structured texts require a different approach than conversations between several speakers, as the latter necessitates considering the interaction to

1. https://fr.wikipedia.org/wiki/Résumé

preserve the most relevant information. The second level of difficulty lies in producing the summary itself, which further highlights the intricacies of automatic text summarization. As stated earlier, there are two approaches for automatic text summarization: extractive text summarization approach and abstractive one. These two approaches are based on the form of the summary to obtain. The extractive approach focuses on selecting the key information to form a summary while the abstractive approach involves generating a condensation of the essential information using paraphrases and new words. Abstractive text summarization is the most challenging approach to perform, but it is the most human-like summarization. Indeed, a deep understanding of the text is assumed to be essential. This understanding allows for the reformulation of the text in a shorter form, using synonyms, paraphrasing, new words, or restructuring sentences. This process is not merely about condensing the text but about capturing its essence, which underscores the complexity and depth of abstractive text summarization. For example, given the following text:

> *SOURCE TEXT: Cristiano Ronaldo is one of the most renowned footballers in history, hailing from Portugal. Known for his remarkable athleticism, skill, and dedication, he has earned numerous awards and accolades throughout his career. Ronaldo's ability to score goals with precision and flair has solidified his status as one of the greatest goal-scorers of all time. Beyond the pitch, he's a global icon, admired for his philanthropy, business ventures, and charismatic persona.*

An extractive summarization can be:

> *EXTRACTIVE SUMMARY: Cristiano Ronaldo, a Portuguese footballer, is recognized for his skill, athleticism, and his goal-scoring ability, earning numerous awards and global admiration throughout his career.*

Here the summary is just an extraction of words from the text. Let compare this summary to the following abstractive summary:

> *ABSTRACTIVE SUMMARY: Cristiano Ronaldo, a Portuguese football sensation, has captivated the world with his extraordinary talent, dazzling skillset, and unparalleled success, cementing his status as a legendary figure in the sport both for his on-field prowess and his off-field impact as a global icon.*

The abstractive summary, as observed, tends to introduce novel lexical items and is predicated upon a comprehensive reconfiguration of the source document content, occasionally accentuated by the amplification of crucial information.

Since it is not clear if current state-of-the-art models are able to perform this deep understanding, the use of different words or paraphrases in automatic text summarization, can cause errors that make the summary inconsistent with the original source document.

An automatic system is a specialized system that is engineered to execute tasks or operations without requiring direct human involvement. When it comes to summarization, the system must be able to produce a short version of a document. To

achieve this, it undergoes training using a collection of Document-Summary pairs, which enables it to identify the features of the document and establish a correlation with the summary. This training helps the system to produce a summary based on an input document. Training a model on abstractive summarization data, which includes both the document and the abstractive summary, could result in the model generating inconsistent information with the source document.

This can happen when elements of the document are misunderstood or when additional information that cannot be inferred from the source is added. These errors are called *hallucinations* in the literature [May+20]. Let's take this example:

> *ABSTRACTIVE SUMMARY HALLUCINATED: Cristiano Ronaldo, the Portuguese football maestro, has conquered the sporting world with his remarkable talent. He has played for some great teams, including Real Madrid and Juventus.*

The source document does not specify whether Ronaldo is a player of Real Madrid or Juventus. Providing this information outside of the document is known as a *factual hallucination*, as the information is accurate yet not explicitly stated within the source material. While this may be clear to those with prior knowledge, it could potentially cause confusion for those without the relevant context. Another example of hallucination is as follows:

> *ABSTRACTIVE SUMMARY HALLUCINATED: Cristiano Ronaldo, the Portuguese football maestro, has conquered the sporting world with his remarkable talent. He has played for some great teams, including Barcelona and scored around 300 goals for his team.*

According to the summary, it is stated that Ronaldo played for Barcelona and scored goals. However, upon closer examination of the source document, this information is not actually included. Those familiar with Ronaldo's journey would recognize that this claim is not accurate. This type of misinformation is referred to as a "*non factual hallucination*," as it is not supported by the source material and is incorrect.

The reason behind hallucinations in text summarization lies in the system's nature. Text summarization systems operate on generative models, which can generate new text that resembles human-like text. These models learn the underlying patterns and structures of language from a vast corpus of text data and then use that knowledge to create new text sequences for various purposes, such as language translation, content creation, chatbots, and text summarization. Generative models are used not only in NLP but also in computer vision, where they must comprehend and produce visual content. At present, generative models are at the forefront of AI, particularly with the advent of Large Language Model(s) (LLM) trained on vast amounts of data. However, this can result in hallucinations while generating text. Indeed, these models are designed to generate fluent human-like text, but not necessarily consistent. Although text summarization systems are intended to produce a summary based on text input, they are not immune to the tendency of generative models to generate information that does not originate from the source document.

The inclusion of extraneous information in a summary not only compromises its fidelity to the original document but also creates confusion. This underscores the urgent necessity for more precise and faithful summarization techniques. In medicine, a summary that accurately reflects the source document is crucial. Given that healthcare providers invest considerable time in summarizing copious amounts of text, including diagnostic reports, progress notes, and a patient's treatment history across multiple specialists, errors in the summary can have significant consequences. Therefore, ensuring the faithfulness of the summary is paramount.

This thesis focuses on the fidelity of generated summaries to their source documents. The importance of this research extends beyond academia, as it has significant implications for industrial applications. Unfaithful summaries can result in severe consequences in an industrial context, emphasizing the urgency and significance of this research. By exploring the fidelity of generated summaries, this research advances the field of automatic summarization and has important implications for the coherence of summaries in an industrial context. This is especially relevant as the thesis is being conducted in an industrial setting at Enedis, where non-coherent summaries can have substantial consequences.

Enedis is a Distribution System Operator (DSO) company that carries out millions of interventions on its electricity network each year. These interventions involve multiple stages, including initial conversations with customers in one of Enedis' call centers and subsequent document creation in various formats, such as tables and texts. Unfortunately, some interventions result in futile trips, which can be challenging to understand and infer the cause. However, Enedis is seeking an automated solution to summarize these interventions and predict the causes of unsuccessful ones, which would be a tremendous asset for the company. In our research, we are dedicated to ensuring the accuracy and faithfulness of these summaries in relation to the source documents, which is a crucial research challenge. This subject holds significance not only for Enedis but also for all entities and systems that employ summarization techniques like those mentioned previously in healthcare domain. It is essential to ensure that the summary generated is reliable. Indeed, what value does it hold if one cannot trust its accuracy? Any inconsistencies in the summary undermine its purpose, which is to provide a comprehensive overview of the text without having to delve into the details. If one cannot depend on the summary, a second reading of the document to verify the information presented in the summary would be necessary, resulting in a time-consuming process.

Given the information outlined above, we have embarked on a research journey to investigate the fidelity of generated summaries to their source documents. This question is of utmost importance to the field of automatic text summarization, which includes general text and conversation transcription summarization. The main question we are exploring in this thesis is:

How can we ensure that the generated summary remains faithful to the

source?

There are two ways of answering this question. First, by explicitly detecting factual errors and try to correct them. Second, by modifying the generation process to prevent factual errors made by the system. For our part, we focus on the second one because avoiding factual errors is a double-edged sword as it also deals with the detection phase.

Thus, to answer this question, we first studied the errors that generally appear in summaries generated by an automatic system using a typology of errors. We also studied the nature of reference summaries and the type of abstraction that can be found in them using a proposed typology. Our investigation has revealed that a significant number of errors in the generated summaries stem from the inclusion of information not present in the source document. As measuring hallucination is difficult without human evaluation, we decided to focus on specific information that can be general for each application of text summarization. Indeed, we choose to study named entities that are inferable for each domain and crucial for text comprehension. This study has paved the way for the formulation of several hypotheses aimed at addressing our research question:

— *Are named entities outside the source document mostly hallucinations?:* We studied the named entities that appear in the generated summary of several systems to test this hypothesis.

— *Will reducing the number of entities outside the source document increase the faithfulness of generated summaries?:* Errors on named entities in systems can be controlled at three levels: upstream of system training (on the corpus), during system training, and during the generation phase. Therefore, we decided to explore two ways of forcing the model to use named entities from the document: during training by changing the objective and during the generation phase by using a selection criterion.

In this study, we focus on named entities as it is easily inferable, but our methodology can be applied to other type of information.

Secondly, this thesis focuses on improving automatic conversation summarization, specifically in task-oriented dialogues. Unlike summarizing written text, dialog summarization involves multiple speakers interacting through asynchronous turns, often with disfluencies and specific roles (such as agent or customer) to accomplish a goal. We wanted to ensure that our generated summaries were faithful to the dialog while maintaining coherence. To achieve this, we formulated several hypotheses:

— The integration of task-related semantic information within the framework of task-oriented dialog summarization will result in a semantically correct summary that is coherent with the dialogue.

— The use of an automatic system to create a new dataset does not negatively impact the fidelity of the summaries generated in relation to the conversation.

— Automatic transcription systems could potentially serve as a viable solution for reducing the cost of the method while maintaining summary coherence.

This thesis presents six key contributions to address various challenges related to hypotheses and try to bring answers to them. These contributions include:

— the proposal of an error typology for automatic summaries and characterization of the abstraction phenomenon on reference summaries,

— the analysis of the risk of hallucination on named entities in text summarization and the proposal of named entities' hallucination risk criterion based on the assumption that out-of-document entities are mostly hallucinations.

— the proposal of a selection criterion to minimize factual errors on named entities in automatic summaries,

— the study of the integration of task-related semantic information in a task-oriented dialog to verify the faithfulness of summaries with respect to dialog,

— the proposal of evaluation criteria based on task-specific information,

— the proposal of a new homogeneous dataset for task-oriented dialog summarization.

These contributions form the backbone of the thesis, with the first chapter (Chapter 1) providing an overview of text and dialog summarization, emphasizing the importance of summary faithfulness to the source document. The second chapter (Chapter 2) delves into an error typology for generated summaries and characterizes the abstraction phenomenon in reference summaries. This analysis serves as a foundation to examine errors made by automatic systems and link them to errors on named entities. Chapter 3 proposes a new criterion to evaluate entities' faithfulness to the source document, aiding models in predicting faithful summaries regarding named entities. Finally, Chapter 4 focuses on task-oriented dialog summarization, proposing the use of task information as a semantic guide to generate faithful summaries, considering semantic measures on the task, such as named entities and call type in case of customer service calls. It also proposed a new dataset for task-oriented dialog summarization.

1 Automatic summarization: State of the art and problematic of the study

Sommaire

1.1 Introduction . . . 34
1.2 Automatic text summarization . . . 36
1.2.1 Text summarization datasets . . . 36
1.2.2 Text summarization Approaches . . . 40
1.2.2.1 RNN/CNN-based Methods . . . 41
1.2.2.2 Transformers-based Methods . . . 46
1.3 Dialog summarization . . . 51
1.3.1 Definitions . . . 51
1.3.2 Dialog dataset for summarization . . . 52
1.3.3 Approaches for dialog summarization . . . 54
1.3.3.1 Meeting summarization . . . 55
1.3.3.2 Chat summarization . . . 56
1.3.3.3 Customer service dialog summarization . . . 57
1.4 Automatic Evaluation measures . . . 57
1.5 Hallucination in automatic summarization . . . 59
1.5.1 Definition . . . 60
1.5.2 Hallucination metrics . . . 61
1.5.2.1 Unsupervised metric . . . 62
1.5.2.2 Weakly Supervised metric . . . 64
1.5.2.3 Others metrics . . . 64
1.5.2.4 Human evaluation of hallucination . . . 64
1.5.2.5 Benchmark for hallucination . . . 65
1.5.3 Methods to relieve hallucination in text summarization . . . 66
1.5.4 Hallucination in dialog summarization . . . 68
1.6 Conclusion . . . 69

1.1 Introduction

This section will provide an overview of the various aspects covered in this thesis, starting with the current state of the art in automatic summarization for text and dialog.

We will also discuss the different metrics used to evaluate automatic summarization in past works, and define the concepts and notions that will be used throughout the following sections. Additionally, we will focus on the unfaithfulness of generated summaries to the source document, and review previous research that proposes methods to evaluate and prevent this issue.

The process of generating summaries can significantly vary depending on several factors:

— *Document Type:* Summaries can be created from various types of documents, such as academic papers, books, news articles, technical reports, etc. Each type of document may require a distinct approach in terms of comprehending content and selecting critical information.

— *Non-Textual Documents:* Some summaries may be derived from non-textual content such as images, audio recordings, or videos. In such instances, multimedia processing techniques can extract relevant information and generate a comprehensive summary.

— *User-Specified Constraints:* Users may request specific requirements for the summary, such as particular topics to be included or excluded, a specific tone, or even a precise summary length. In such cases, summary generation systems need to consider these constraints while still producing an informative and coherent summary.

— *Headline or Synopsis Generation:* The generation of headlines or synopses differs from that of the entire summary in terms of length and level of detail. Headlines are usually brief, whereas synopses can provide a more detailed yet concise overview of the content. For instance, a title can summarize an article in a few essential words, while a synopsis can summarize each paragraph or section in a few sentences.

— *Summary Compression:* Generating a compressed summary involves significantly reducing the content while maintaining essential information. For example, a 30% compressed summary may involve selecting the most critical sentences and removing less relevant details while preserving the summary's overall coherence and comprehensibility. Compression can sometimes cause essential information to be lost if the compression percentage is too small compared to the amount of information to be represented.

Generating summaries can vary based on various elements such as document type, user requirements, desired summary length, and the degree of compression required. Specific approaches tailored to each use case are often necessary.

Let's describe automatic text summarization, the dataset and the different approaches.

1.2 Automatic text summarization

Automatic text summarization is a task of NLP. It is the process of automatically generating a summary of a document by capturing its key information. There are two methods of automatic summarization: extractive summarization and abstractive summarization. Extractive summarization involves extracting the most important information from the document and then concatenating it to form a summary. Abstractive summarization involves paraphrasing. It consists in creating a summary of a text using paraphrases or new words, while retaining the most salient information in the document. As it requires a prior understanding of the document before being summarized, abstractive summarization is the most human-like summarization. It is therefore more difficult than the extractive summarization. Figure 1.1 gives an example of abstractive and extractive summarization.

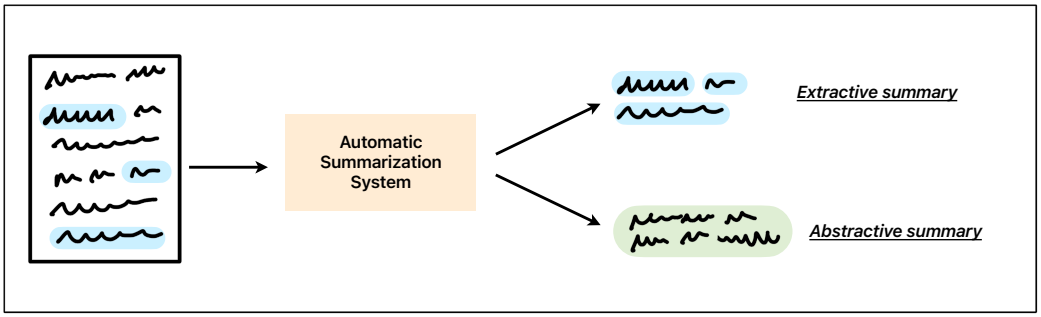


Figure 1.1 – Example : Extractive vs Abstractive summarization

Since the study focuses on abstractive summarization, we will present in this section the state-of-the-art datasets and techniques introduced for abstractive text summarization.

1.2.1 Text summarization datasets

Obtaining a dataset for automatic text summarization has historically been a challenging task. To effectively train an automatic text summarization system, it is essential to have both the original text and its corresponding summary. For the purpose of text summarization, two distinct types of datasets were developed. The first ones were created specifically for this task, with a meticulous attention to detail and annotation guidelines such as DUC dataset. On the other hand, the second type comprised of "opportunistic" datasets that were collected because they existed, but without a prior plan for harmonization or thoughtful annotation by the annotators such as XSum, CNN/DM. There exist a greater number of opportunistic corpora as compared to those specifically designed for text summarization tasks, complete with meticulously annotated summaries. The creation of a corpus comprising of manually annotated

summaries entails significant costs. Therefore, to further progress in this area, opportunistic corpora were introduced and subsequently embraced. Here we will explore the different datasets introduced for the task based on these two type.

One of the first datasets was built for Document Understanding Conferences (DUC), an international contest that aimed to encourage advances in the field of NLP [HO04; Dan06]. This annual contest ran from 2000 to 2007 and provided competitors with dataset intended solely for the evaluation of summarization methods. The first two DUC datasets (DUC 01, DUC 02) were single-document summaries, while the subsequent ones (DUC 2003, DUC 2004) were drawn from several data sources, including the New York Times and the Associated Press Wire services. For DUC 2004, each document had 4 human summaries associated with it. The different corpora provided for the DUC task were not enough to train text summarization models.

One of the first corpora introduced for the abstractive text summarization task was Gigaword [Gra+03; RCW15]. It consists of 4M samples extracted from news articles sourced from seven different publishers within the Gigaword corpus[Gra+03]. The goal of the task on Gigaword was to generate headlines based on the content of the first sentence of each article. Thus, the summary associated with each document in this corpus does not concern the document in its entirety, which limits the task of automatic text summarization. To fill the gap, the CNN/Daily Mail corpus was introduced [Her+15; Nal+16]. It consists of 311K news articles from the CNN and the Daily Mail websites. The summaries are multi-sentence summaries from the bullet point summaries created by the publishers. This was followed by several dataset proposals using news articles available on the Internet. [GNA18] proposed NEWSROOM, while [NCL18] proposed XSum. NEWSROOM includes 1.3 million pairs of articles and summaries, written by NEWSROOM professionals in 38 leading publications between 1998 and 2017. This makes it a dataset that has summaries with a diversity of writing styles. NEWSROOM summaries have been created according to different strategies. There are extractive, abstractive and mixed summaries. As for XSum, it contains 227,000 British Broadcasting Corporation (BBC) articles published between 2010 and 2017, spanning a broad range of topics such as Politics, News, Education, Sports, Technology, Science, Health, Family, Entertainment, Weather, Business and Arts [NCL18]. These articles were accompanied by professionally written one-sentence summaries. There's also the New York Times (NY Times) dataset, that was introduced for abstractive summarization by [San08]. It was then used for the first time in an abstractive summarization by [PXS18]. It is a corpus composed of 600K pairs of articles and abstracts written by a staff of library scientists. [Fab+19] proposed Multi-News, a multi-document summarization dataset. Multi-News consists in 56k pairs (articles-summaries) and is the first large-scale news datasets for multi-document summarization. The articles come from the website newer.com and are associated with corresponding human-written summaries.

These various datasets were in a general news domain and did not always reflect future industrial uses of the methods. [Coh+18] proposed PubMed and arXiv, two corpora from scientific repositories (PubMed.com for PubMed and arXiv.org for arXiv) which enabled to work on two problems in the case of a closed domain. The first is the length of the document to be summarized, and the second is the use of technical terms from the domain. PubMed and arXiv are made of 133K and 215K articles respectively, and have an average length of 3000 and 5000 words per article, whereas news articles average around 500 words per article. In addition, scientific articles have been methodologically written, with supporting summaries.

Many others introduced text summarization datasets for specific use-case. [KW18] proposed WikiHow, a text summarization datasets that consists of 200k pairs extracted from the WikiHow.com website. It is composed of multiple instruction-step paragraphs with a summarizing sentence. A summarization dataset on U.S. patents, BIGPATENT [SLW19], was also introduced. BIGPATENT is made of 1.3 million examples. Each example consists in a summary written by a human and a category representing the type of patent to which it belongs. Another corpus that has been introduced is AESLC [ZT19]. AESLC is an e-mail dataset for text summarization that consists of 18K e-mails. The e-mails used to create the dataset come from the Enron corpus [KY04], which consists of e-mails from Enron corporation employees. The aim is to summarize e-mails. Here, e-mail subjects are considered as e-mail summaries. The problem with this corpus is that the subject of an e-mail does not necessarily reflect its content. Datasets based on online discussion forums have also been introduced. Reddit TIFU [KKK19] is a corpus that was created from informal stories posted on the Reddit forum. The data was collected on TIFU subreddit and comes from posts from 2013 to 2018. It consists of 120K posts associated with either short summaries (Title of the post) or long summaries (TL;DR). The particularity of this corpus is that the posts are informal and they are written following rules to be posted on this subreddit. Each post must be written with a title and a long summary (TL;DR) associated.

Most of the corpora introduced for research in the field of automatic summarization are in English. As a result, researches in other languages do not benefit from this advance. To fill the gap, several multilingual corpora for automatic summarization have been introduced, including MLSum [Sci+20] and XL-Sum [Has+21]. MLSum is the first large-scale summarization dataset that contains more than 1.5M pairs (article-summary) in six different languages (English, French, German, Spanish, Russian, Turkish). [Sci+20] collected the data from different news websites according to language from 2010 to 2019 under conditions: news should be on general topics, have humans written highlights/summaries and contain a wide range of available articles. For English, articles from the CNN/Daily Mail datasets were taken [Nal+16]. In addition, they obtained a dataset composed of 400k of French, and about 300k for the other languages except Russian, which is in reduced quantity (about 30k). XL-Sum [Has+21],

on the other hand, is a more recent multilingual dataset for abstractive summarization based on news articles. It consists of more than 1M pairs (articles-summaries) and covers 44 languages. [Has+21] picked up articles from the BBC news website, which has articles written in 43 languages. Unlike XSum [NCL18], they retrieved the article summary written by the authors at the beginning of the article to summarize it. This summary can be one or two sentences long. Using human evaluation of their corpus, they showed that the majority of summaries (90%) covered the entirety of the associated articles. However, one-third of the summaries contained information that was not mentioned in the source document (i.e. could not be inferred from it).

Datasets	**#doc**	**ADW**	**ADS**	**ASW**	**ASS**	**% novel n-grams**		**ABS**	**CMP**
						n=1	**n=2**		
Gigaword	3804K	30	1	8	1.00	41.94	80.26	0.411	0.720
CNN/DM	287K	701	39	51	3.68	17.87	55.96	0.153	0.910
NY Times*	655K	822	34	46	1.00	22.64	55.59	0.249	0.869
NEWSROOM*	995K	677	26	40	1.75	-	-	0.191	0.910
XSum	203K	392	19	21	1.00	37.36	83.92	0.333	0.906
arXiv	215K	5860	252	267	10.97	9.54	41.23	0.094	0.558
PubMed	133K	3044	100	204	7.5	14.47	46.66	0.143	0.884
AESLC	14K	130	7	4	1.02	41.79	66.26	0.409	0.949
Reddit title	80K	319	18	7	1.00	35.54	74.95	0.333	0.954
TL;DR	42K	400	22	21	1.45	31.85	78.39	0.286	0.917

Table 1.1 – Statistics on abstractive text summarization dataset. ADW - Average Document Word, ADS - Average Document Sentence, ASW - Average Summary Word, ASS - Average Summary Sentence. The statistics of dataset with * was taken from the paper [BC20] Other statistics were generated by us.

We reported in Table 1.1 statistics (size, number of words and sentences in the summary and the document) of some of the datasets presented in this section. We also computed the degree of abstraction of these corpus. To measure abstractiveness, [NCL18] proposed using the percentage of novel n-grams in the summary (i.e. n-grams that do not appear in the input document). [GNA18] proposed to use *fragments* to match spans shared between the source document and the summary which [BC20] used to quantify abstractivity. They defined abstractivity as a normalized function of the combined length of fragments. They additionally suggested utilizing compression as a metric for measuring conciseness.

Table 1.1 shows the different measures presented previously for some corpora. Regarding the results, XSum is the most abstractive dataset for the task of automatic text summarization as the percentage of novel n-grams (n=1, n=2) is higher than the others.

Note 1.1: Degree of abstractivity

Let D and S be the source document and the summary associated respectively, and $\mathscr{F}(D,S)$ the *fragments* set between D and S. Abstractivity (ABS) and compression CMP will be computed as follow:

$$ABS(D,S) = 1 - \frac{\sum_{f \in \mathscr{F}(D,S)} |f|}{|S|} \tag{1.1}$$

$$CMP(D,S) = 1 - \frac{|S|}{|D|} \tag{1.2}$$

As previously mentioned, certain datasets, such as XSum and CNN/DM, are opportunistic dataset for summarization. However, these datasets do pose some challenges in that their summaries may not always fulfill the intended purpose of summarization, which is to encapsulate the most significant information within a document. For instance, XSum consists of one-sentence summaries that may not necessarily contain any relevant information from the original text, resulting in higher percentages of novel n-grams (n=1, n=2) and ABS. Similarly, CNN/DM relies on news article headlines that are designed to grab readers' attention, rather than serve as a comprehensive summary. As a result, the use of these datasets for automatic summarization tasks raises questions regarding their suitability given the way they were created.

1.2.2 Text summarization Approaches

In general, Machine-learning approaches to text generation involve creating text automatically based on computer models trained on large quantities of text data. These approaches can be divided into two main categories:

— *Rule-based language models:* These models use linguistic rules and probability statistics to generate text. Markov chains and n-gram models are common examples. The probability of a word sequence is calculated as a function of the frequency of sequences observed in the training corpus.

— *Neural language models:* These models use neural networks to learn the structure of natural language and generate text. Recurrent Neural Network (RNN), Long-Short Term Memory (LSTM) and transformers are widely used. They are capable of capturing long-term dependencies in text.

These models are trained on vast amounts of text data to learn the structure and style of natural language. Once they are trained, they can be used to generate text for various applications like machine translation, summary generation, automatic article writing, and many others.

Earlier works on automatic text summarization were based on statistical methods [KM02; BMW00]. These methods were introduced for the task of extractive text summarization. Unfortunately, the texts selected and combined to form the summary suffered from consistency problems. There were several problems with pronoun resolution (co-reference) and repetition. Several methods were developed to correct the

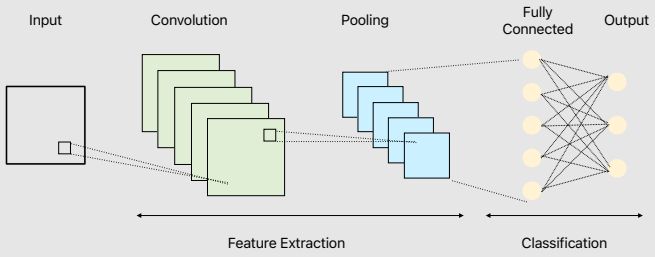
various issues, but there was still room for improvement.
On the other hand, abstractive text summarization was challenging. Therefore, the work was focused mainly on extractive summarization. Some authors proposed to perform abstractive summarization task by filling in predefined patterns [SL00; Oya+14; Tri17]. Others turned to extractive summarization methods to correct the various problems and modify specific sentences. Despite their efforts, they were not able to produce an abstractive summary. It wasn't until the advent of the Deep Neural Networks (DNN), in particular the RNN and Convolutional Neural Network (CNN), that the task of abstractive text summarization gained renewed interest. Several works based on RNN and CNN followed until the arrival of the Transformers architecture [Vas+17], which revolutionized the NLP by its efficiency on several tasks, in particular abstractive text summarization.

In this section we focus on the methods proposed for abstractive text summarization task, based on the RNN/CNN and Transformers architecture.

1.2.2.1 RNN/CNN-based Methods

Note 1.2: More details about CNN ?

CNN or Convolutional Neural Network is a type of Deep Neural Network DNN that is extensively used in tasks such as image recognition, computer vision, and object detection to analyze and process visual data. It is composed of convolutional layers which extract features from images and learn representations hierarchically, making it a powerful tool for recognizing patterns in visual data.



In NLP, one-dimensional convolutions are commonly used to capture local patterns and relationships among words or sequences of words in text.

Attention-Based Summarization (Abs) [RCW15]. The contribution of the model proposed by [RCW15] involves both, a neural probabilistic language model and an encoder that provides a conditional summarization model, which was separate in previous work. By following the work done in machine translation, the original distribution, a conditional language model based on the input sentence, is parameterized directly as a neural network. [RCW15] estimates the contextual probability of the next token generated using a feed-forward Neural Network Language Model (NNLM) to adapt the language model. To include the input text into the generation, they proposed testing three different encoders as follow:

— Bag-of-Words Encoder: The input sentence was represented as a bag-of-words which was embedded down to a specific size. Despite its benefits in catching the relevance of each word, this model has several limitations, particularly when it comes to representing a sentence. This representation does not take into account the context of each word. Indeed, it ignores relationships and orders.

— Convolutional Encoder: It was introduced to fix some bag-of-words' issues. Indeed, the convolutional encoder, using a CNN, allows a local interaction between words.

— Attention-based Encoder: Based on works done in machine translation, [RCW15] used a contextual encoder based on attention to have a representation based on the generation context.

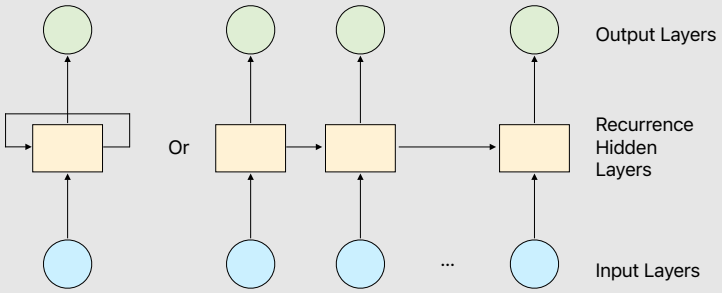
Although their model gave accurate abstractive summaries, the encoder was limited by context. Indeed, it allows only a local interaction i.e. only the immediate context was captured.

Attentive Recurrent Architecture (s2s-att) [CAR16]. The authors introduced a conditional RNN which, given an input, generates a summary. Their method is an extension of the model proposed by [RCW15]. [CAR16] replaced the feed-foward NNLM decoder used by [RCW15] with a RNN and used the convolutional attention-based encoder. By explicitly encoding the positional information of the input words, [CAR16] suggested an improved version of the one proposed by [RCW15]. At each time step, the encoder produced scores over the words of the input sentence, which help the decoder to focus on a specific part of the input sentence to generate the next word. Thus, it overcomes the limitations of their model by capturing a broader context.
For the conditional RNN decoder, [CAR16] decided to take a look at two different types of RNN : LSTM [HS97] and Elman RNN [Elm90].

Sequence-to-sequence RNNs (s2s-Rnn) [Nal+16]. Their first suggestion was to use an attentional encoder-decoder RNN which had originally been introduced for machine translation [BCB16]. Secondly, they addressed specific problems of text summarization by introducing other models. The baseline model based on attentional encoder-decoder RNN consists on a bidirectional GRU-RNN [Chu+14] for the encoder and a uni-directional GRU-RNN for the decoder with the same size and an attention mechanism over the source-hidden states. To generate words, a soft-max layer over the target vocabulary was used. They also used a technique call Large Vocabulary Trick (LVT) [Jea+15] that consists in reducing the softmax layer computation and thus reducing training costs. The decoder's vocabulary for each mini-batch is limited to words found in the source documents of that batch; to which they added the most frequent target words to obtain a fixed size vocabulary.

Note 1.3: More details about RNN ?

RNN is a deep neural network that operate sequentially. It processes input sequence token by token to encode and decode. Unlike traditional neural network, RNN can use information from previous inputs to make predictions. The main characteristic of a RNN is its ability to remember past inputs over time.



Traditional RNNs have certain limitations when it comes to capturing long-term dependencies in sequences. This issue is commonly referred to as the "vanishing" or "exploding" gradient problem, where the information may gradually decrease or increase exponentially over many time steps. This makes it difficult for the network to learn and remember long-range dependencies. To solve this problem, several RNN architectures were introduced such as LSTM [HS97] and Gated Recurrent Unit (GRU) [Chu+14]. These architectures include mechanisms, such as gates, to regulate information flow, mitigate the vanishing gradient problem, and capture longer-term dependencies in sequences more effectively than standard RNNs. LSTM consists of 3 gates (input, forget, and output) while maintaining a separate cell state and hidden state. Whereas the GRU has 2 gates (update=input+forget, reset) and combines cell state and hidden state. This makes the LSTM more complex, with more parameters than the GRU, and therefore able to capture more patterns than the GRU. On the other hand, because of its architectural simplicity, the GRU is less expensive to compute.

[Nal+16] addressed three major issues in text summarization:

— Identifying key concepts and entities: To address this issue, they proposed to use a word embedding-based representation of the source document. It was concatenated to one embedding vector each for Part Of Speech (POS), named entities, and discretized Term Frequency (TF) and Inverse Document Frequency (IDF) values.

— Dealing with Rare/Unseen Words: Since the input document frequently contains words unknown to the vocabulary, they proposed using the pointer/generator mechanism to tackle this problem. During the decoding time, the decoder can decide, at every time-step, to generate a word according to its normal function, or to point to the location of a word in the source document, by using a switch function.

— Identifying key sentences: To capture the key sentences from which the summary

can be formed, two bidirectional RNNs are used. The bidirectional RNNs, located at the source side, take into account the sentence level and the word level. Thus, the model will consider the context and the importance of sentences in the document.

[Nal+16] is close to the work done by [RCW15] and [CAR16] but with some major differences: they used an RNN-based encoder decoder, with a pointer/generator mechanism coupled to a hierarchical attention for a decoder. Also, adding features such as POS, Named Entity Recognition (NER) tags, and TF/IDF values is a plus. The problem of [Nal+16]'s method is that it only summarizes short input sequences and has an attentional encoder-decoder model that generates unnatural summaries consisting of repeated sentences.

Summarization with Pointer-Generator Networks (PtGen) [SLM17]. They addressed issues on the use of sequence to sequence model such as the inaccurate reproduction of factual detail, the problem of Out-Of-Vocabulary (OOV) words and the repetition in the generated summary. They proposed 3 models :

— The baseline model: That is a sequence-to-sequence attentional model with a single-layer bidirectional LSTM as encoder and a single-layer unidirectional LSTM as decoder. They used probability distribution as attention to know where to look to produce the next word.

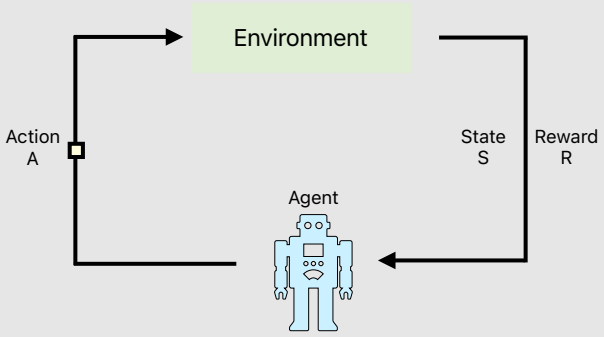
— Pointer-generator model: This solve the problem of OOV words. A generation probability was used to choose between generating word from the vocabulary and copying word from input sequence. The generation probability enables to raise or to low the probability of all generated words, or all copied words at once. The main difference between their switch (pointer/generator) and the one made by [Nal+16] is that their model learns when to use the pointer.

— Pointer-generator + Coverage mechanism: It allows to avoid repetition. It replaces the temporal attention mechanism used by [Nal+16]. Their coverage approach is the same as [Xu+16] who apply it to image-captioning. The coverage vector is the sum of the attention distributions of all previous decoder timesteps.

Although pointer-generator methods have focused on solving the previous problems of automatic summarization, such as OOV words and repetition, there's still room for improvement. Thus, others proposed methods based on Reinforcement Learning (RL).

Note 1.4: More details about RL ?

RL stands by Reinforcement Learning is a machine learning branch focused on training agent to make decisions in an environment to achieve specific goal. Maximizing a cumulative reward is the agent's aim.



Let's define some terms related to RL:

— The environment refers to the external system or scenario wherein the agent interacts.

— A policy is the strategy adopted by the agent to determine its actions in a given state.

— Actions refer to the choices available to the agent in a specific state.

— Reward is the feedback provided to the agent after it takes an action in a specific state.

Deep RL model (DeepRL) [PXS18]. They proposed an intra-attention model based on the encoder-decoder network [SVL14]. This addressed the problem of repetition and the inefficiency of previous models, like the one of [Nal+16], on long input sequences. Therefore, they presented an intra-attention model with a new training objective based on RL. [PXS18] suggested using a hybrid learning objective as training objective. It consists in a "teacher forcing" training objective [WZ89] that minimizes a maximum-likelihood loss (cross-entropy loss) at each decoding step, combined to a self-critical policy gradient training objective [Ren+17]. Since discrete metrics such as ROUGE do not always yield the best results when minimizing maximum likelihood, learning a policy that maximizes a specific discrete metric is a solution. Therefore, they used a reward function resulting from the comparison of the generated sequence and the ground truth summary according to a selected evaluation metric.
Another reason why [PXS18] decided to use the learning policy is to reduce exposure bias. Indeed, a "teacher forcing" training assumes that ground truth is provided at every step during training. This leads to an accumulation of errors because there is no information about the ground truth at the time of inference.
As far as the attention mechanism is concerned, the main difference with previous work in automatic summarization lies in the intra-decoder attention, which allows the

model to look at what has been generated before at each step in order to avoid repetition. The system also incorporates the pointer mechanism used by [Nal+16]. A human evaluation of the generated summaries shows that their mixed-objective proposal performs better in terms of readability and relevance than separate objectives.

Fast-abs-rl [CB18]. The models discussed previously aimed to increase text fluency through an attention model. However, these models can take a long time to decode a word because, during this phase, it must look at all the encoded words to decode each word in the generated summary. Also, while many have worked to reduce repetition in the generated summary, it remained challenging. Thus, [CB18] proposed Reinforce-Selected Sentence Rewriting, which is a hybrid extractive-abstractive architecture with policy-based RL. Their model first selects the salient sentences in the source document using an extractor agent, and then rewrites the selected sentences to abstract them using an abstractor network. The extractor agent is based on RL and pointer networks. It used a ROUGE [Lin04] reward to maximize the maximum likelihood training and extract sentences based on their representations learnt by a hierarchical network. The abstractor is an encoder-aligner-decoder that enables compression and paraphrasing of sentences extracted using the extractor. It is coupled with a copy mechanism described above to avoid OOV [SLM17]. Despite the use of an extractor, their method gives good abstractivity results compared with the previous one. In fact, the ratio of new n-grams in the generated summary is higher than those obtained previously. However, their system has a problem with the metric as a reward for the RL. Indeed, RL's reward is a sentence-level ROUGE while the metric for evaluating generated summaries is a summary-level ROUGE.

Bottom-Up [GDR18] Neural abstractive models described above produce more fluent text but are poor at content selection. Training models using RL can be challenging to fine-tune and often requires lengthy training times. [GDR18] proposed Bottom-Up, a neural abstractive model that makes a selection mask for the source document and then generates the summary from this selected mask. The goal here is to ascertain the sentences that need to be added to a summary. The selection process is structured as a sequence-tagging task, where the primary goal is to identify pertinent tokens within the source document. As far as the neural model is concerned, it has the same architecture as the model introduced by [SLM17].

Let's explore transformer-based methods after RNN/CNN methods.

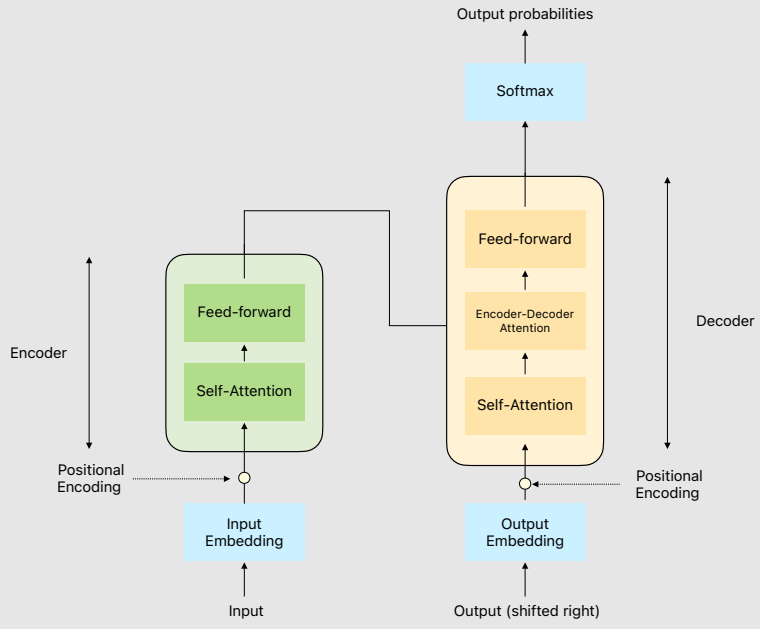
1.2.2.2 Transformers-based Methods

Here, we will present automatic text summarization methods based on the Transformers architecture introduced by [Vas+17]. Transformers is a current state-of-art architecture used in natural language processing that outperforms previous models on almost all NLP tasks. The most important addition of Transformers architecture is the self-attention mechanism, which allows the model to weight the importance of different parts of the input sequence when processing it.

For the methods based on Transformers, the models are first pre-trained on large corpus and then finetuned on specific data under an objective to perform text summarization tasks.

Note 1.5: More details about Transformers architecture ?

Transformers is a neural network architecture introduced by [Vas+17]. It allows to capture relationships between different elements in a sequence, enabling effective modeling of long-range dependencies in data. Transformers consists in an *encoder* that processes the input sequence, and a *decoder* that generates the output sequence. Each part (encoder/decoder) have multiple layers of *attention mechanisms* and *feed-forward neural networks*. The attention mechanism of Transformers is a multi-head attention instead of single attention mechanism. It allows the model to capture different aspects and relationships within the input data. A *positional encoding* is added to input embeddings to provide information about token position in sequence since transformers do not inherently account for sequential order of input tokens. We do not describe in details Transformers architecture here but you can refer to [Vas+17] for more details.



BERTSUM [LL19]. It is an automatic text summarization method based on Bidirectional Encoder Representations from Transformers (BERT). BERTSUM used the pre-trained encoder of BERT and a randomly initialized decoder based on the transformer architecture introduced by [Vas+17] for the task of automatic text summarization. Instead of using one [CLS] token to represent the entire document, [LL19] used multiple [CLS] tokens placed at the start of every sentence within a document and interval segmentation embeddings in the process. Their approach is divided into two

stages. First, the encoder is trained for an extractive summarization task. Then, it is also trained for an abstractive summarization task. Its ability to express the semantics of a document and obtain representations for its sentences enabled it to beat the systems presented above.

Note 1.6: More details about BERT ?

Introduced by [Dev+19], BERT is a multi-layer bidirectional Transformer encoder based on the Transformers architecture [Vas+17] with various number of layers. BERT-base contains 12 encoder layers while BERT-large contains 24 encoder layers. BERT was trained with two objectives: Masked Language Model (MLM) and Next Sentence Prediction (NSP). The pre-trained MLM objective consists in randomly masking some tokens in the input sentence. The goal is to predict these masked words with respect to the context of the sentence. The NSP objective allows the model to capture and understand the relationship between two sentences. It consists in a binarized task: predicting whether the sentence is the next one.

BART [Lew+20]. BART is a pre-trained denoising autoencoder Transformer-based model. BART's pre-training consists first in applying an arbitrary noising function to corrupt the text deliberately, and then training a model to reconstruct the original text from this corrupted version. What's special about BART is that it consists in a bidirectional encoder like BERT and an auto-regressive decoder. These components enable it not only to generate a token from previous tokens (auto-regressive setting), but also to process the entire input sequence at once (bidirectional setting). They used 5 ways to corrupt the text (example: Figure 1.2):

— Token Masking which consists in sampling random tokens and replacing them with [MASK] elements as BERT did.

— Text Deletion which consists in deleting a random token in the input. The model decides which positions of the input text must be deleted.

— Text Infilling which allows training the model to predict the number of missing tokens from a span. Thus, several spans chosen using a Poisson distribution are replaced with a [MASK] token.

— Sentence Permutation which consists in shuffling, in random order, the input sentences.

— For Document Rotation, the idea is to train the model to identify the document's start. Thus, a token, chosen uniformly, is rotated to the beginning of the document.

BART has two architectures: the Base architecture which contains 6 encoder layers and 6 decoder layers, and the Large architecture with 12 encoder and decoder layers. BART's architecture makes it effective for various text generation tasks, including automatic text summarization. It gave better results on the XSum [NCL18] and CNN/DM [Nal+16] corpus than BERTSUM, which is also based on Transformers.

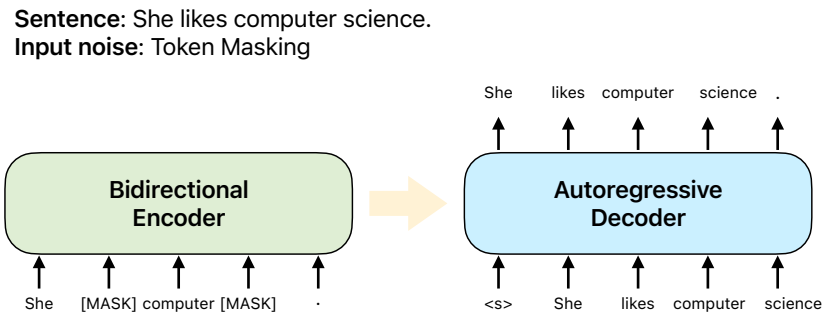


Figure 1.2 – BART architecture [Lew+20]

T5 [Raf+20]. Text-To-Text Transfer Transformer (T5), is a pre-trained language model designed as text-to-text framework for different tasks. Unlike model presented previously, every natural language processing task (translation, summarization, ...) is cast into a text generation problem. T5 was trained to generate a target text corresponding to a task prompt in the input text. For example, to perform summarization, the input document is preceded by the prefix "summarize". T5 used mostly the same architecture as the original Transformer [Vas+17] and was pre-trained with the same objective as BART (randomly corrupted spans) on their dataset Colossal Clean Crawled Corpus (C4). C4 was made using text from the Common Crawl's website, a publicly-available web archive. Common Crawl provides web text by removing non-text content. [Raf+20] cleaned the Common Crawl's web extracted text using different heuristics to address the problem of not having natural language text. C4 is a dataset of about 750GB of English text introduced for the pre-training. For text summarization task, [Raf+20] showed that T5 gave good results on CNN/DM [Nal+16] corpus by generating coherent summaries.

PEGASUS [Zha+20a]. It is a pre-trained Transformer-based encoder-decoder model trained following two different objectives tailored simultaneously for abstractive summarization (Figure 1.3).

— Gap Sentences Generation (GSG). They replaced important/principal sentences in the input document by a special token [MASK1]. This sentence is used as a pseudo-summary of the document. Thus, the idea is to generate this masked sentence. This allows to have the pre-training objective close to the text summarization task. To select the number of gap sentences, they proposed using 3 strategies: *Random* for selecting uniformly and randomly a number of sentences, *Lead* that consists in selecting the first sentences, and *Principal* that selects top-m sentences according to importance gave by a metric. Among the 3, *Principal* was the one that gave the best result for abstractive summarization tasks.

— MLM. As BART, they followed BERT to select 15% of tokens in the input text that

should be masked. Then 80% of the time, they replaced the selected tokens by the token [MASK2], 10% of the time by a random token, and 10% of the time they left the token unchanged.

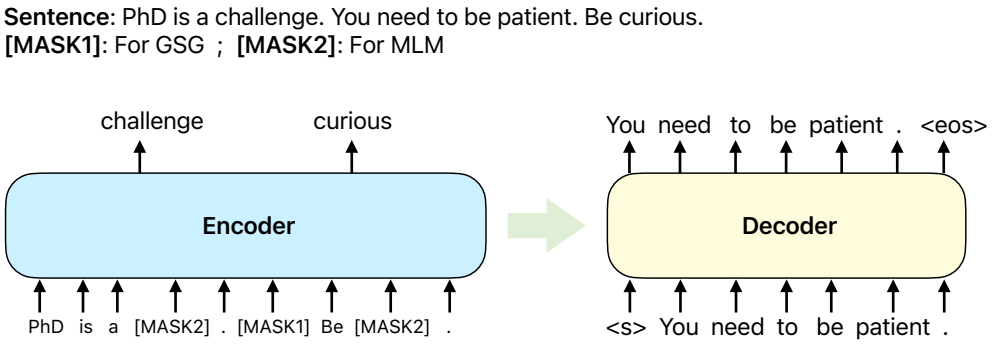


Figure 1.3 – PEGASUS architecture [Zha+20a]

PEGASUS was pre-trained on two large text corpora: C4 [Raf+20] (presented before) and HugeNews. HugeNews consists of 1.5 Billions articles from news and news-like websites. The data was collected from 2013 to 2019. [Zha+20a] fine-tuned PEGASUS on 12 abstractive summarization datasets including CNN/DailyMail [Nal+16] and XSum [NCL18] and obtained state-of-the-art results on most of the dataset. They observed qualitative results in terms of fluency and coherence from their generated summaries with regard to the ground truth. To evaluate manually the quality of the summaries generated, they conducted a human evaluation (rating different summaries on a 1-5 scale) of the summary on 3 datasets including CNN/DailyMail [Nal+16] and XSum [NCL18] and obtained significant results.

For each model presented, we report in Table 1.2 the various datasets they used to test their model.

Transformer-based models have revolutionized the task of automatic summarization, but a significant problem still needs to be solved: the unfaithfulness of generated summaries to the source document. Indeed, these summaries contain information that do not come from the source document. The models make associations between the data on which they have been pre-trained and the corpora on which they have been refined, which sometimes introduces inconsistencies. We will address this matter with greater depth at a later time, but first, let's talk about dialog summarization, as it is another focus of this thesis.

	Abs	s2s-att	s2s-Rnn	PtGen	DeepRL	Fast-abs-rl	BERTSUM	BART	T5	PEGASUS
DUC-2004	×	×	×	-	-	×	-	-	-	-
Gigaword	×	×	×	-	-	-	-	-	-	×
CNN/DM	-	-	×	×	×	×	×	×	×	×
NY Times	-	-	-	-	×	-	×	-	-	-
NEWSROOM	-	-	-	-	-	-	-	-	-	×
XSum	-	-	-	-	-	-	×	×	-	×
arXiv	-	-	-	-	-	-	-	-	-	×
PubMed	-	-	-	-	-	-	-	-	-	×
AESLC	-	-	-	-	-	-	-	-	-	×
BIGPATENT	-	-	-	-	-	-	-	-	-	×
Reddit TIFU	-	-	-	-	-	-	-	-	-	×

Table 1.2 – Model - datasets

1.3 Dialog summarization

1.3.1 Definitions

Before defining dialog summarization, we focus on dialog. Dialog refers to a conversation or interaction between two or more participants, typically involving alternating exchanges of *messages* or *utterances*. It can be spoken or written interactions. The participants are commonly named *speaker*. An utterance refers to individual spoken or written expressions made by speakers during the conversation, often in the form of sentences or phrases. Speakers interact with each other using turn-taking. He is said to be taking a *turn*. A *turn* refers to the sequences of speech or text (in the case of written dialog) produced by a speaker before handing over the conversational floor to another speaker. An example of a dialog is given in figure 1.4.

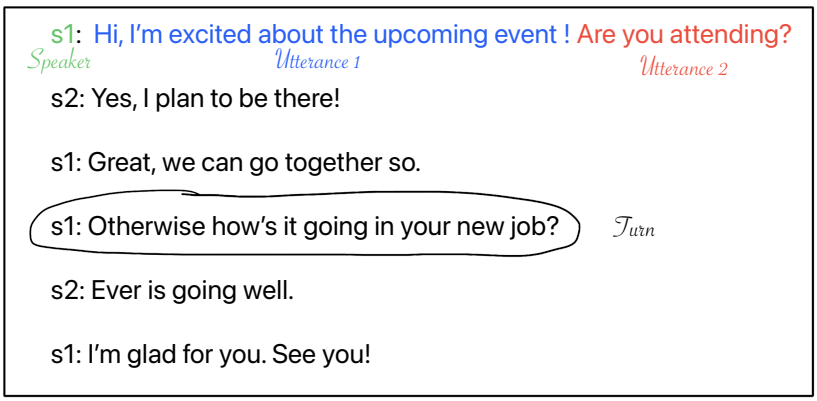


Figure 1.4 – Example of dialog

Dialog can include exchanges in discussion forums, interactions between a human and a machine, or telephone discussions and meetings between several people. This variability of dialogue leads to a variability in the roles of those involved. Thus, dialog can be divided into two types: Open-Domain Dialog (ODD) and Task-Oriented Dialog (TOD).

— ODD: The topics are open-ended, and the roles of the speakers may fluctuate or even disappear throughout the discussion. It involves discussions on diverse and unrestricted topics.

— TOD: It is a type of conversation where the primary goal is to accomplish a specific task or objective. The task at hand demands precision, and every role is clearly delineated with no room for ambiguity. Customer service interactions, technical support, and similar scenarios often involve these types of dialogues.

Now that we know what a dialog is, we can describe automatic dialog summarization. Dialog summarization is an extension of text summarization to spoken or written interactions. It is based on conversations involving multiple *speakers* interacting over asynchronous *turns*. The style of spontaneous speech found in transcripts does not match the expected style of the summaries. This entails that dialog summarization is best tackled with abstractive approaches than extractive ones. Here we will present past works on dialog summarization as well as several datasets made available for the task.

1.3.2 Dialog dataset for summarization

The problem of annotated corpus for text summarization was also a problem for dialog summarization. Several dataset types for automatic dialog summarization have been introduced.

The first corpora introduced was AMI/ICSI [Car+05; Jan+03]. AMI/ICSI are both English corpora based on meeting conversations between many speakers. It was introduced not only for dialog summarization but also for audio and video recordings. The Augmented Multi-party Interaction (AMI) project gathered data for its corpus by recording 137 meetings with durations ranging from 15 to 45 minutes. The meetings were scripted with the aim of simulating a work environment, where four participants played specific roles - a project manager, a marketing expert, a user interface designer, and an industrial designer - for a fictitious company. The participants were expected to communicate naturally and spontaneously during the meetings. The ICSI corpus [Jan+03] consists of 75 one-hour meetings with an average of six participants. The participants, who are members of a research team at the International Computer Science Institute, are recorded while discussing topics related to their research.

Meeting corpus differs from dialogue in that the objectives are not the same. Also, AMI/ICSI being a small corpus did not allow for the advancement of research on automatic dialogue summarization. Thus, [Gli+19] proposed SAMSum, a corpus

whose aim is to study conversations exchanged on social networks such as Messenger and WhatsApp. It is a chit-chat dialog corpus which contains written conversations created by linguists fluent in English. The aim was to reproduce messenger-style messages, i.e. with several levels of formality, containing slang phrases, emoticons and typos. SAMSum consists of 16k online chats with corresponding human-written summaries annotated by experts. The SAMSum's summaries include interlocutors' names. Therefore, it introduced a new challenge that consists in modelling the dialog and the participants. One of the problems with the SAMSum corpus is that it is derived from written chat. It, therefore, has fairly short dialogues and a language that differs from real-life spoken dialog since it can contain emojis, abbreviations, etc. For a more real-life application of spoken dialog summarization, [Che+21b] proposed DialogSum, Real-Life Scenario Dialogue Summarization that consists of 13,460 conversations and associated summaries. DialogSum is composed of dialog data from Dailydialog [Li+17], DREAM [Sun+19], and MuTual [Cui+20], three public dialogue corpora. Annotators wrote the summaries of DialogSum, and human evaluation was performed to ensure good-quality summaries. Expired by the lack of diverse multi-speaker conversation summarization datasets in SAMSum, [KZS21] proposed ForumSum, a conversation summarization dataset collected from internet forums. They took annotators to create the summaries associated with each conversation. To ensure the quality of the summaries, they annotated theses summaries on a scale of 1 to 5 and asked annotators to re-annotate those that didn't score well. What makes ForumSum so special is the fact that conversations take place between an average of 6 participants on a wide variety of subjects.

Another type of corpus for dialog summarization that has be introduced is a customer service dataset. Customer service is crucial for shaping consumer experiences. It significantly affects customer satisfaction, retention, and brand perception. Providing prompt, empathetic assistance and tailored solutions can build customer trust and loyalty. A summary of the conversation between the customer and the agent helps improve the service offered. Customer service involves two types of participants in a task-oriented dialogue that covers various domains and contains informative entities. [Bec+12] introduced DECODA [1], a french dialog summarization dataset, for this task. It contains real-life conversations between RATP users and telephone advisors. DECODA's conversations are from a call center and have been fully anonymized. This corpus consists of more than 1,500 conversation transcripts and corresponding synopses (small summaries) presenting the main events of the conversation: caller goal, resolution status... Each dialog takes place between potentially multiple agents from the call center and a caller (the user). [Zha+21] introduced TODSum and [Lin+21] introduced CSDS two task-oriented dialog summary corpora. TODSum is a large-scale dataset based on MultiWOZ [Bud+18], the largest human conversational corpus that consists of 10,438 samples and covers seven different domains. To enhance the quality of the summary, they consider the structure dialog state. TODSum's summaries were

1. We will provide more information on this corpus as it will be used for our experiments.

built using a trained generation model that generated candidate summaries from the conversation. Then, a human evaluation and correction are performed to improve the generated summaries. This process was repeated several times to obtain a perfectly annotated corpus. TODSum consists of 9,906 conversations with associated summaries and dialog states. CSDS [Lin+21] is a Chinese role-oriented summarization dataset that was introduced to capture various speakers' perspectives. Each summary of CSDS focuses on individual topics, maintaining the dialog's structure at the topic level. Thus, each conversation is associated with the topic, the summary from the agent's perspective, the summary from the customer's perspective and the overall summary of the conversation.

We reported in Table 1.3 information about the dialog summarization datasets presented before. #dialog is the size of each corpus (number of conversations), #spk is the average number of speakers for each dialog, #turn is the average number of turns for each dialog, #conv.len and #sum.len are the average conversation and summary length in term of words respectively, and Lang is the language of the conversation (EN: English, FR: French, CH: Chinese).

Datasets	Domain	# dialog	# spk	# turn	#conv.len	#sum.len	Lang
AMI	Meeting	137	4	535.6	6000.7	296.6	EN
ICSI		75	6	-	-	-	EN
SAMSum	Chat	16,369	2.40	9.9	93.79	20.30	EN
ForumSum	Forum	4,058	6.73	10.13	303.45	35.95	EN
DECODA	Customer Service	1,514	2	-	-	-	FR
TODSum		9,906	-	14.1	186.9	45.4	EN
CSDS		9,101	-	26.0	-	-	CH
DialogSum	Spoken	13,640	-	9.5	131	22.6	EN

Table 1.3 – Dialog summarization datasets: Statistics, domains and languages

1.3.3 Approaches for dialog summarization

The first methods introduced for automatic dialogue summarization were based on extractive methods, which involved selecting the most relevant utterances for summarization. However, given the structure of the dialog, extractive methods were not suitable. Relevant information is scattered throughout the conversation, and may be incoherent at times. With the evolution of abstractive text summarization methods thanks to the advent of neural networks, the task of automatic dialogue summarization has become affordable. In this section, we present some state-of-the-art methods for automatic dialog summarization.

1.3.3.1 Meeting summarization

[Sha+18] introduced the first work for abstractive dialog summarization. He proposed a straightforward approach for compressing multiple sentences to create a summary of meetings datasets such as AMI/ICSI [Car+05; Jan+03]. Their proposed method is composed of 4 different modules. The conversation is pre-processed in the first module. The second step is Community Detection, which groups together related statements discussed in a meeting. An abstractive sentence is generated for each community using the Multi-Sentence Compression Graph (MSCG) [Fil10]. A summary is then created by selecting the best elements from the set of abstractive sentences. The advantage of their method is that it has no supervision (annotated data). Dialogues are different from text because of their structure based on exchanges between two and more participants. Thus, traditional methods for text summarization did not give good results on text summarization. Therefore, many research focused on using information about the dialog or dialog structure to perform dialog summarization. While [GC18] proposed used dialog acts, [Koa+20] using domain terminologies.

Dialog act was commonly referred to the impact of a spoken statement on a conversation (eg. questions, statements, requests for action). The main idea of [GC18] was to use the interactive behavior of the dialog to perform summarization. By introducing a novel sentence gate, the dialog act signal can be conditioned on the learned summarization result, leading to better performance.

We have discussed the problems that long summary documents can introduce in automatic text summarization approaches. This problem is all the more critical in automatic conversation summarization. In fact, links need to be made to the various utterances of the conversation despite the length of the dialogue. [Zhu+20] introduced HMNet, a dialog summarization system based on a hierarchical modeling strategy to deal with long meeting transcriptions. It is based on the Transformers architecture [Vas+17] pre-trained on news summarization corpora that have been converted into the meeting format. HMNet included a role vector representing the speaker's role, which was appended to the turn-level representation and used for decoding to accommodate multi-speaker scenario.

On the other hand, [Zho+21a] introduced query-based summarization, which focuses on specific aspects like decisions, actions, ideas, or hypotheses. Based on the assumption that meeting summaries do not always need to cover everything, their proposed model summarizes particular parts of meetings based on provided queries. They located the relevant text according to the query using a hierarchical ranking-based model as a "Locator". Then, they generated a summary from existing models such as PTGen [SLM17], BART [Lew+20] ou encore HMNet [Zhu+20] by giving them the query and the relevant text spans as input. Although they obtained good results using the Locator in terms of the usual measure for evaluating automatic summaries,

they noted that 31% of summaries generated by BART are completely unrelated to the given query.

1.3.3.2 Chat summarization

The challenge of the task of chat summarization is to model the speaker and the interaction in the summary. Thus, many works were proposed to face this challenge.

[CY20] suggested a sequence-to-sequence multi-view dialog summarization system with topic segments and conversational stages. They used BART-large [Lew+20] as the pre-trained model for their multi-view dialog summarization model. They studied the challenges and emphasized the importance of dialog modelling for summarization. The analysis of different errors in the generated dialog showed that their best model suffers mostly from referral/coreference errors and role/language change. This analysis also showed that the model encountered difficulties in recognizing essential information. The disadvantage of methods based on dialog information is that the information is extracted using toolkits not designed for dialog. For example, the model proposed by [CY20] depends on topic segmentation. [Fen+21] proposed an unsupervised method for annotating dialog to bridge the gap. It allows the extraction of essential information, topic segmentation, and redundancy detection.

[ZXG20] modelled the dialog using topic words in a dialog graph attention network. This allows them to capture to capture long-distance inter-sentence relations. They introduced Topic-word Guided Dialogue Graph Attention (TGDGA), a Graph Neural Network (GNN) to identify intra-sentence and inter-sentence relations and generate summaries through a graph-to-sequence framework and topic words. According to human evaluation, the summaries generated by TGDGA contain more salient information than the previous methods and are grammatically correct.

On his side, [LC21] introduced a planning personal named entities to model dialog participants. The idea of their method was to guide dialogue summarization with personal named entity planning. A transform-based model was trained to generate a summary by taking a personal named entity planning as input. This planning is created by intersecting the personal named entity sets from the source document and the reference summary. The resulting set is called Occurrence Planning and was ordered by the personal named entity's appearance in the source document. The model generates a summary based on document content and a list of named entities as input. Two planning methods are used for inference: Comprehensive Planning covers all information, and Focus Planning provides speaker-specific summaries. The model was evaluated on SAMSum [Gli+19] using standard measures such as ROUGE, achieving better performance. A human evaluation of the consistency and informativeness of the generated summaries showed that Comprehensible Planning achieves a better informativeness score than Occurrence Planning but with a slightly lower consistency score.

Despite advances in automatic chat summarization, challenges remain. They still need room for improvement because selecting the essential parts of the dialog is difficult, and the models tend to generate inconsistent information about the input dialogue.

1.3.3.3 Customer service dialog summarization

Customer service conversation summarization is an important industrial application for improving customer service. The proposed models must consider the task's major challenges: modelling the speakers (agent/customer), the evolution chains, and the inherent topics. A shared task to improve dialog summarization were introduced by [Fav+15]. It is Call Centre Conversation Summarization (CCCS) that involves generating textual synopses of spoken conversations that can inform on their content and facilitate browsing a database of recordings.

[TFB16] proposed a method for creating summaries using pattern filling. The method involves identifying important concepts within a conversation and then filling a pre-defined pattern with that information. First, the pattern to be used is detected, and then it is filled in with the extracted information for each conversation. This method cannot be considered as an abstractive summarization method but represents a first brick for customer service dialog summarization.

According to [Liu+19a], an auxiliary task can be used to perform customer service dialog summarization. They suggested generating a sequence of key points, such as Question description ⇒ Solution ⇒ User approval ⇒ End, to indicate the logic of the dialog and then summarizing it. They defined 51 key points that was generated for each summary using pre-processing rules and showed that generated the sequence of key points before the summary improves the dialog summary.

[Zou+21] developed a novel mechanism for multi-role topic modelling. They employed neural topic modelling to explore topics implicitly. This mechanism is designed to address the issue of insufficient data. On the other hand, [Zha+20b] proposed an unsupervised framework called RankAE, which is aimed at producing more accurate summaries. This framework selects topic utterances based on their centrality and diversity, which improves the quality of the summaries. The final summaries are generated using a denoising auto-encoder, which eliminates noise from the input and produces clear and concise summaries.

1.4 Automatic Evaluation measures

In this section, we will present the automatic metric used to evaluate the generated summaries of a summarization system.

ROUGE [Lin04]. ROUGE stands for Recall-Oriented Understudy for Gisting Evaluation. It is the most widely used measure for evaluating automatic summarization systems. There are different types of ROUGE:

— ROUGE-N: It assesses the model's ability to generate n-grams close to the reference summary. An n-gram refers to a combination of tokens or words in a sequence. A unigram (1-gram) comprises a single word, while a bigram (2-gram) consists of two consecutive words.

$$\text{ROUGE-N} = \frac{\sum_{s \in S_{\text{ref}}} \sum_{\text{ng} \in s} \text{C(ng)}}{\sum_{s \in S_{\text{ref}}} \sum_{\text{ng}} N(ng)} \tag{1.3}$$

— ROUGE-L: It computes the longest common sub-sequence between the reference summary and the generated summary with the assumption that the longer the shared sequence, the more similarity between sequences.

$$R_{lcs} = \frac{LCS(\text{hyp}, \text{ref})}{\sum_{s \in S_{\text{ref}}} \sum_{\text{ng}} N(ng)} \qquad P_{lcs} = \frac{LCS(\text{hyp}, \text{ref})}{\sum_{s \in S_{\text{hyp}}} \sum_{\text{ng}} N(ng)} \tag{1.4}$$

$$F1_{lcs} = \frac{(1-\beta^2) \times P_{lcs} \times R_{lcs}}{R_{lcs} + \beta^2 P_{lcs}} \tag{1.5}$$

One drawback of ROUGE score is its inability to capture unfaithful information regarding the source document. Indeed, for abstractive summarization, ROUGE score is not widely appropriate to evaluate a summary as the summary can be a paraphrase of the reference summary.

METEOR [BL05]. Metric for Evaluation of Translation with Explicit Ordering, also known as METEOR, is a widely used metric for assessing the quality of machine translations. It calculates the harmonic mean between the precision and recall of a sentence based on unigrams, with a greater emphasis on recall than precision. While initially designed for machine translation, METEOR can also be used for automated summarization, as it has been found to correlate with human judgment.

To calculate this metric, an algorithm must be followed. First, a word-level alignment between the two sentences is created. In the case of automatic text summarization, this is the reference summary sentence and the generated summary sentence. This alignment must respect specific rules, such as that a word can only be aligned a maximum of once. Then, Precision P and Recall R at word level, and the harmonic mean between Precision and Recall F_{mean} are computed. For m is the number of unigrams in common between the hypothesis and the reference, w_t, w_r the number of unigrams of the hypothesis and the reference respectively. P, R and F_{mean} are computed as follows:

$$P = \frac{m}{w_t} \qquad R = \frac{m}{w_r} \qquad F_{mean} = \frac{10PR}{P+9R} \tag{1.6}$$

Measures introduced so far only assess single-word congruity, not larger segments shared by both sentences. Longer n-gram matches are used to calculate a penalty score that increases with the number of non-adjacent mappings. For N_w the number of words, and N_{au}, the number of aligned unigrams, the penalty score is shown below:

$$p = 0.5 \times (\frac{N_w}{N_{au}})^3 \tag{1.7}$$

The METEOR Score is then computed as follow:

$$M = F_{mean}(1-p) \tag{1.8}$$

Pyramid [NPM07] There are various ways to communicate an idea, such as "Alice invites her close friends over for dinner, She requests them to join her for a meal". However, evaluating these expressions automatically can be challenging. To address this issue, [NPM07] proposed a semi-automatic method called Pyramid. This method involves identifying semantic units (Summarization Content Unit (SCU)) by analyzing reference summaries created by humans. SCUs that describe the same objects or actions are combined and given a weight based on their frequency in the reference summaries. Based on this weight, a pyramid is constructed with the most frequent SCUs at the top and the least frequent at the bottom. The summary is then scored based on the number of semantic units that are considered important by the annotators (i.e., those at the top of the pyramid). However, identifying and aligning SCUs between the system-generated summaries and the reference summaries is a manual process, which can be time-consuming and costly.

BERTScore [Zha+20c]. BERTScore is a similarity measure based on contextual word embedding (BERT) applied to the token sequence of reference and candidate summaries. It compares the generated summary with the reference summary and assigns a score based on how similar they are. The goal of this computation is to maximize the cosine similarity between contextualized token embeddings. The BERTScore was introduced to overcome the difficulties of measures based on n-gram overlap such as METEOR or ROUGE in evaluating paraphrases which is a characteristics of abstractive text summarization. Assuming the following sentence[2] as reference: *"people like foreign cars"* and the two sentences as hypotheses *s1: people like visiting places abroad* and *s2: consumers prefer imported cars*, the usual measures based on n-gram overlap will give a higher score (relative to the reference) to sentence s1 than to sentence s2.

1.5 Hallucination in automatic summarization

Neural networks have achieved remarkable results especially those with the Transformer architecture. However, these systems generate summaries that contain incon-

2. The example was taken from the referred paper [Zha+20c]

sistent information regard to the source document. [Cao+18] have shown that 30% of the summaries produced by a selection of automatic text summarization systems, notably those proposed by [RCW15] and [CAR16], contained inconsistencies. And [Fal+19] indicate that cutting-edge abstractive summarization systems, assessed using metrics like ROUGE [Lin04], BLEU [Pap+02], and METEOR [BL05] , have included inconsistent information in 25% of their summaries. This inconsistent information is not in the source document. As these inconsistencies are recurrent in automatic text generation and particularly in automatic summarization, the name *hallucination* was introduced to refer to them [May+20]. [May+20] have demonstrated that when it comes to summarization, abstractive models have a tendency to generate information that may not be entirely accurate to the source document. Despite this, [Zho+21b] have found that these unfaithful summaries can have high ROUGE scores even with these inaccuracies.

What are hallucinations, how can a summary be evaluated in relation to them, these are the different points that will be addressed in this section.

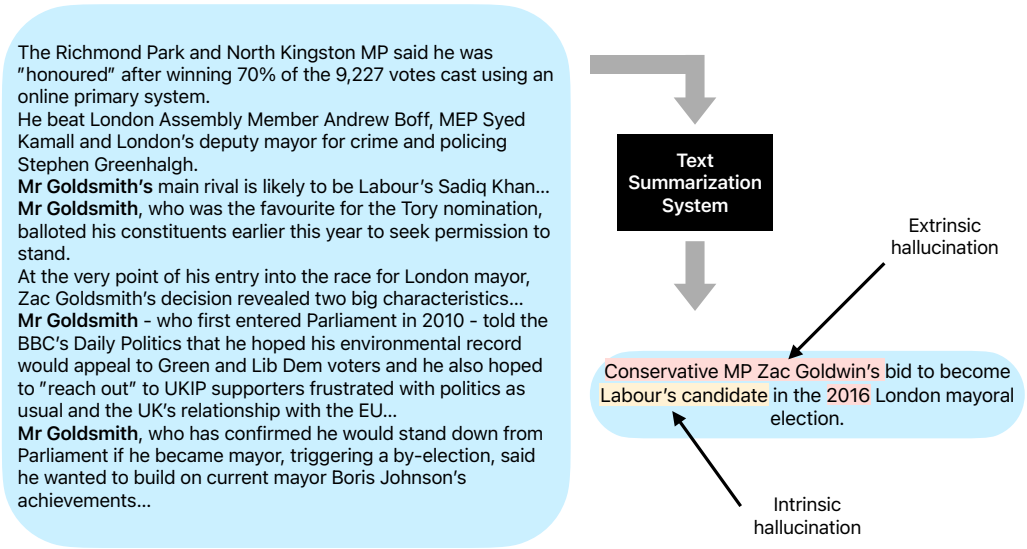


Figure 1.5 – Example of extrinsic and intrinsic hallucination. [May+20]

1.5.1 Definition

The term *hallucination* was first introduced by [May+20]. They define *hallucination* as the summary content unfaithful to the input document. By conducting human evaluation of different systems outputs, they proposed two types of hallucination: Intrinsic Hallucination (Int-Hall) and Extrinsic Hallucination (Ext-Hall).

— Int-Hall is information from the document that is not consistent with it. According to them, Int-Hall may be due to a misrepresentation of the source document

by the models.

— Ext-Hall is defined as summary information that is outside the source document. A poorly-informed decoder and a model that ignores divergence issues between source and target texts can lead to Ext-Hall.

Note 1.7: Formal Definition of Intrinsic and Extrinsic Hallucination

Let D, S and $I = i_1 i_2 i_3$ be a document, a summary associated and an information from the summary respectively.
I is said to be an Int-Hall if the terms and concepts that make up information I are in the source document but information I is not.
$\Rightarrow i_1 \in D, i_2 \in D, i_3 \in D$ but $I \notin D$.
I is said to be an Ext-Hall if the terms that make up information I are not in D.
$\Rightarrow i_1 \notin D, i_2 \notin D, i_3 \notin D$ and $I \notin D$.

Figure 1.5 gives an example of intrinsic and extrinsic hallucination to understand these two notions better. Problems of inconsistency between information in the summary and that in the source document have led to the introduction of a number of terminologies that should not be confused: *hallucination*, *faithfulness* and *factuality*. while faithfulness refers to maintaining unwavering consistency and truthfulness in relation to the given source, *factuality* denotes the quality of being genuine or grounded in verifiable facts. In some cases, the definition of factuality is confused with faithfulness when the input document is chosen as the source to determine the factual correctness. This is what [Don+20] do. On the other hand, [May+20] consider the source of facts to be the world knowledge.

[PBT21] proposed specific and detailed categories for factual errors that occur within summaries. They introduced three main categories of errors: content verifiable error, discourse error, and semantic frame error. These fact errors can be considered as hallucination according to the typology of [May+20] since the source of fact-finding is limited to the source document.

Hallucinations have given rise to two directions. Some focus on the evaluation of summaries in terms of faithfulness, while others propose methods to reduce or even avoid them. The following sections will focus on these different points.

1.5.2 Hallucination metrics

Different type of metrics was proposed to access summary faithfulness. Some methods are unsupervised, which means they don't rely on pre-labeled data, while others are weakly supervised and use a combination of labeled and partially labeled data. For unsupervised metrics, the works that will be presented in this section are based on Natural Lnaguage Generation (NLI) and Question Answering (QA) models. Weakly supervised metrics aim to reduce the cost of data labeling required for a faithful evaluation by eliminating the need for a large quantity of annotated data. These metrics are often based on a modification of the data.

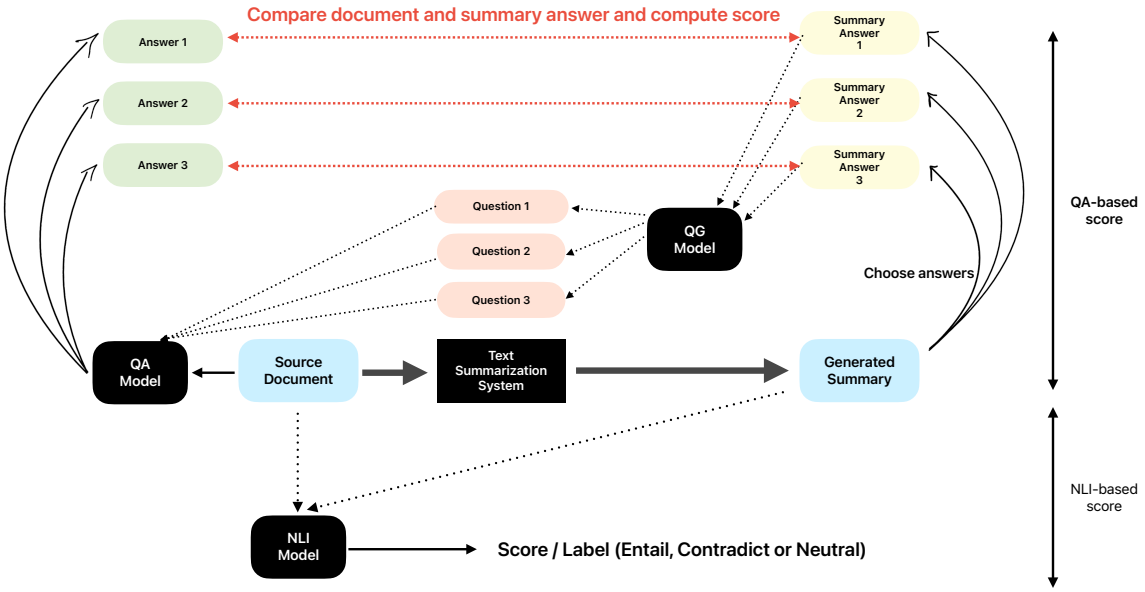


Figure 1.6 – Framework of QA and NLI based metrics

1.5.2.1 Unsupervised metric

NLI-based metrics Based on the hypothesis that *a faithful summary must entail the source document content*, many proposed methods based on NLI named also textual entailment to evaluate a generated summary. The method aims to determine if a summary entails or contradicts the source document as shows Figure 1.6. [Fal+19] investigated this avenue but were unable to obtain conclusive results on the evaluation of hallucinations. Indeed, the NLI models used could not be transferred to the automatic summarization task due to their training data. Further studies were carried out on the use of NLI for the evaluation of hallucinations in summarization. [May+20] finetuned a BERT-large pre-trained model on Multi-NLI [KKK18] for entailment classification to evaluate the generated summaries of different systems on the XSum corpus [NCL18]. They showed that the entailment score obtained was moderately correlated with faithfulness and weakly correlated factuality with human evaluation, while the measures usually used (ROUGE and BERTScore) had a very weak correlation. On his side, [Mis+21] proposed the creation of a dataset suitable for summary evaluation using NLI methods. Indeed, they found that the commonly used NLI methods didn't work well enough on summaries due to the different data types. Traditional NLI methods are trained on shorter texts than in an automatic summarization document. Using data augmentation techniques, they automatically converted multiple-choice reading comprehension datasets into two-class NLI datasets. This enabled them to achieve better results than previous models for NLI. More recent works proposed was SUMMAC [Lab+22]. It is an NLI-based method that applies CNN on entailment score between document and summary sentences to access summary faithfulness.

[Lab+22] showed that their proposed technique outperforms NLI models on the entire document.

QA-based metrics Many studies have been conducted to assess the quality of generated summaries using QA models. The underlying assumption is that *if the summaries are factually consistent with the source materials, the QA models should provide similar answers to questions.* As part of their review process, they cross-checked the answers in the document obtained using Question Answering model against the answers of generated summary. The QA-based metrics consist of four steps as shows Figure 1.6:

1. Extract information from the generated summary which will be targeted answer. Information can be n-grams named entities or Noun Phrase (NP).
2. Generate questions from answers extracted from the summary generated using a Question Generation (QG) system.
3. Using the source document as context, answer the question with a QA model.
4. Compute the score between the response obtained from the document's QA system and that extracted from the generated summary using an alignment metric.

The first models introduced were QAGS and FEQA [WCL20; DHD20]. Then, [Sci+21] introduced QuestEval and [Nan+21b], QUALS. More recent work that proposed QA-based metric for faithful evaluation was QAFactEval [Fab+22]. Let's take a closer look at these QA-based metrics to highlight the differences. QAGS, FEQA and QuestEval are mostly the same. They used the same QA system (based on BERT), QG (based on BART) and metrics to compared with the different answers (F1 score). The difference between QAGS and FEQA are on the extracted information (FEQA used named entities as extracted information, QAGS used n-grams). One of the new measures added to QuestEval is designed to generate a question from the original document and answer it using the generated summary, instead of the traditional approach of asking a question from the summary and answering it using the document. [Nan+21b] proposed combining the model of QA and QG into a single system QUALS capable of generating a question and its answer. According to the research findings, reference-free metrics significantly correlate with human judgments regarding faithfulness compared to the baseline metrics. A study conducted by [Gab+21] revealed that the effectiveness of QA-based metrics depends on the question being asked. Furthermore, the researchers propose a meta-evaluation framework that integrates QA metrics. When it comes to evaluating question-answering models, [Fab+22] has suggested a new metric called QAFactEval. The metric follows a similar four-step process like other QA-based metrics, but with a few differences. [Fab+22]'s analysis revealed that selecting models and extracting information are key factors in maximizing the performance of QA-based models. To extract information from the summary, they used the concept of NP. In contrast to FEQA and QAGS, they utilized a BART-based QG model [DGL18] and filtered the generated questions to only keep the best ones.

For the QA model and metric, Electra-large [Cla+20] was used as the model, while Learned Evaluation metric for Reading Comprehension (LERC) [Che+20] was used as the metric to compare the two responses.

1.5.2.2 Weakly Supervised metric

FactCC is a weakly supervised NLI-based model proposed by [Kry+20] for evaluating factual consistency. The model is trained jointly for three tasks: predicting whether each sentence in the summary is factually consistent or not, extracting a span from the source document that confirms the consistency if it exists, and extracting the non-coherent span for each sentence in the summary that is not coherent. They showed that the model gave better results than previous proposals.

[Zho+21b], for its part, proposed a model for hallucination detection that was trained on synthetic data for the task generated using BART [Lew+20]. The idea is to predict for each generated summary whether each token is non-true to the source document. This approach enables the measurement of the number of hallucination tokens in a summary, which was not possible with previous methods.

1.5.2.3 Others metrics

[Nan+21a] proposed new metrics (precision-source, precision-target, recall-target) for quantifying hallucinations on named entities. The metric is built on the assumption that *if there is any hallucination, the gold and generated summary will have a different set of named entities.* For $\mathcal{N}(t)$ and $\mathcal{N}(h)$ the number of named-entities in the reference summary and the generated summary respectively, for $\mathcal{N}(h \cap s)$ and $\mathcal{N}(h \cap t)$ the number of named-entities in the generated summary that are also in the source document and the number of named-entities in the generated summary that are also in the reference summary, the precision-source is $\text{prec}_s = \mathcal{N}(h \cap s)/\mathcal{N}(h)$, the precision-target score is $\text{prec}_t = \mathcal{N}(h \cap t)/\mathcal{N}(h)$ and the recall is $\text{recall}_t = \mathcal{N}(h \cap t)/\mathcal{N}(t)$. They also compute a F1 Score based on the precision-target and the recall-target: $F1_t = 2 \cdot \text{prec}_t \cdot \text{recall}_t / (\text{prec}_t + \text{recall}_t)$.

[YNL21] proposed BARTScore, a method for evaluating generated text using pre-trained models BART [Lew+20]. It measures the probability of text being generated from or generating other inputs and outputs. BARTScore proposes a metric with several variants that can be applied flexibly to evaluate text from different perspectives such as informativeness, fluency, or factuality. To assess faithfulness, the authors suggest computing the probability $p(y|x,\theta)$ from the source document to the hypothesis. This measure determines the likelihood of the hypothesis y being generated based on the source text x.

1.5.2.4 Human evaluation of hallucination

Assessing hallucinations is still a difficult task despite the existence of several metrics. As a result, researchers are increasingly relying on human evaluations to gain greater insight into this phenomenon. Techniques for assessing hallucinations by humans can

be broadly categorized into two groups. The first involves comparing the generated summary with the reference summary in terms of fluency, adequacy/relevance, coherence or consistency. In contrast, the second group of techniques does not rely on a reference summary but instead assesses the degree of hallucination or the consistency of the generated summary on a scale. Table 1.4 gives a description of these human evaluation factors.

Factor	Description
Fluency	It is used to measure the coherence and grammaticality of the sentence in summary.
Relevance	It measures the number of important contents carried from the document.
Consistency	It allows to measure the factual correctness of a summary.
Coherence	It measures the quality and the structures of sentences in the summary.
Informativeness	It refers to how effectively the generated text captures the essential ideas of the source text.

Table 1.4 – Human evaluation factors

1.5.2.5 Benchmark for hallucination

More and more research is being carried out to create systems with information faithful to the source document. A rigorous evaluation is therefore needed. With this in mind, [Lab+22] have proposed a Benchmark containing various data for system evaluation. This benchmark comprises a consistent set of reference data and consistency judgments. So, they standardized the six most extensive datasets for abstract inconsistency detection into a binary classification task (label: consistent or inconsistent). The dataset that have been standardized are given in Table 1.5. Most of these dataset were generated from model trained on the CNN/DM dataset [Nal+16]. Only two datasets were based on XSum [NCL18].

Dataset	Description	Size	% Positive	Data source
CoGenSum	Introduced for inconsistency detection. Summaries were labelled as consistent/inconsistent.	1681	49.8	CNN/DM
XSumFaith	Annotated corpus in terms of extrinsic or intrinsic hallucination	2500	10.2	XSum
Polytope	The summary was annotated with eight possible errors, with a severity level for the error.	1268	6.6	CNN/DM
FactCC	Inconsistency detection dataset. Synthetic data were made for negative sample.	1434	85.0	CNN/DM
SummEval	The summary was labeled using a 5-point Likert scale in terms of relevance, coherence, consistency, fluency	1700	90.6	CNN/DM
FRANK	The summary was annotated following seven error types gathered into 3 categories (semantic frame, discourse and content verifiability errors).	2246	33.2	CNN/DM + XSum

Table 1.5 – SUMMAC Benchmark [Lab+22]. Statistics of the different dataset : CoGenSum - Correctness of Generated Summaries [Fal+19], XSumFaith - eXtreme Summarization Faithfulness [May+20], Polytope [Hua+20], FactCC [Kry+20], SumEval [Fab+21], FRANK [PBT21]

Given a consistent and inconsistent summary for a common document, a ranking model should score the consistent summary higher.

1.5.3 Methods to relieve hallucination in text summarization

Different types of methods were introduced to mitigate hallucination in summary generated. Some are based on information extracted from the source document to guide the model during generation. Others proposed using auxiliary tasks search as QA, NLI or RL. Here we will present these different methods.

[Li+18] proposed a method based on textual entailment. It was an abstractive text summarization approach which incorporated entailment knowledge into the model. They used a multi-task learning framework and a reinforcement learning (RL) framework with entailment rewards. While decoding, the model considers the entailment score as a reward combined with a maximum likelihood training process through RL. [Cao+18] conditioned the generation of summary by the source text and a fact description extracted from the source document to reduce hallucination. They used a dual-attention decoder to ensure that the model generated faithful summary. By evaluating the model manually, they showed that their model reduce by 80% the unfaithful summaries on Gigaword [RCW15].

[Fal+19] proposed a re-ranking method based on textual entailment. Generated summaries were raked by their entailment score obtained on the source text. Their proposed approach is based on the assumption that more faithful summary must entail the source document. Unfortunately, the method was limited by the NLI method used. [ZCW20] proposed a method to reduce numeral hallucinations in generated summaries. They do this by re-ranking candidate summaries based on the verification score of quantity entities. They called the system HERMAN. It recognizes quantities like dates and amounts of money in generated summaries and verifies their factual consistency with the source text. Then, during inference time, the most faithful summary based on the quantity hallucination score is selected to ensure that the summary aligns with the source text in terms of quantity.

[Che+21a] proposed a post-processing technique for preventing hallucinations in summary generation. Their approach involves generating several candidate summaries and replacing named entities with those found in the original document. Then, a contrastive candidate selection model selects the best candidate as the final output summary. This method ensures that the generated summary is more accurate and representative of the original text. [CW21] also suggested using a contrastive learning model called CLIFF to improve faithfulness. Based on common errors found in BART and PEGASUS, they designed strategies to create unfaithful summaries. Unlike [Che+21a], which used a contrastive model to select the best candidate from a set of candidates, [CW21] proposed to integrate it during training as an objective in order to help the model to distinguish between inconsistent summaries and reference summaries.

[Nar+21] proposed FROST a pre-trained objective that stand for FReeze entity-level infOrmation in abstractive SummarizaTion with planning. Their model generate summaries based on the entity chains present in the reference summary during training. FROST allowed to mitigate hallucinations at entity-level in a generated summary. Indeed, the idea of this pre-trained objective was to generate an entity chain before generating the summary to allow the model to be prompted by an entity chain that is wanted in the generated summary. FROST enable to drop out hallucinated entities by giving a content plan without hallucinated entities to the decoder. This encourages the model to generate faithful summaries. They showed that their approach achieved state-of-art results on different datasets such as XSum [NCL18], CNN/DM [Nal+16] and SAMSum [Gli+19]. They also showed that modifying the entity chain prompts can reduce hallucination in the generated summary.

[Nan+21a] proposed a multi-task model that predicts whether an entity is important for summary generation or not to reduce hallucination on named entities. They have implemented a loss function that depends on two components: one loss for summary generation and another for named entities prediction in the form of BIO (Begin, Inside, Outside). [Zhu+21] proposed FASUM, a model for extracting and integrating factual

relationships during the summary generation process using graphs, and FC, a model for automatically correcting erroneous facts in summaries generated by different systems. FASUM is based on a seq2seq Transformers architecture that consists in an encoder and decoder part. The decoder's attention is coupled to a graph attention network obtained by feeding, into a BiLSTM model, a transformation applies on the information extraction. The FC model utilized a seq2seq Transformers architecture initialized by RoBERTa-Large weights. Given a document and a candidate summary, the purpose of the model was to generate a more accurate summary with minimal modifications. The authors showed that the model reduces factual errors on CNN/DM as annotators prefer their model in a side-by-side comparison of factual consistency.

Although there are various models available that can help to reduce inconsistencies in the summaries generated, there is still a significant problem of hallucinations in automatic summarization. Therefore, the only way to ensure a summary that accurately reflects the source document is through human intervention during human evaluation.

In the next section, we will talk about hallucinations in the context of automatic dialog summarization.

1.5.4 Hallucination in dialog summarization

Here, we will discuss hallucinations in dialog summarization and how to evaluate them. [HCL22] analyzed different metrics such as ROUGE [Lin04], BERTScore [Zha+20c] and BARTScore [YNL21] on dialogue summarization and showed that most of them have low correlations with human judgements in term of faithfulness. On his side, [Wan+22] show that 35% of summaries generated by different systems on dialog summarization datasets such as SAMSum dataset [Gli+19]. [Tan+22] conducted a comprehensive examination of factual errors in dialog summarization, categorizing them into eight distinct error types (*Missing Information, Redundant Information, Circumstantial Error, Negation Error, Object Error, Tense Error and Modality Error*), among which five are specific to dialog summaries. These errors cannot be used to categorize so-called extrinsic hallucinations (the model imagines information not present in the dialogue). [Wan+22] proposed six types of errors more suitable to evaluate hallucination in dialog summarization. Among them, three types overlap with those proposed by [Tan+22]. These are *Subject Object Error, Pronoun Error and Negation Error*. The news types introduced were *Particulars Errors, Hallucination Error and Other Error*. *Particulars Errors* is an error that occurs when the summary presents inaccurate or false details despite being relevant, while *Hallucination Error* occurs when the summary includes events that were never presented in the conversation.

Fewer methods have been proposed to evaluate and reduce the hallucinations of conversation summarization systems. [Wan+22] proposed a model-level evaluation schema to access faithfulness using a summarization model based on conditional

generation probabilities to distinguish positive and negative summaries generated for dialog using back-translation or rules. [Tan+22] used a *Contrastive Loss* and a *Self-supervised Loss* in addition to the cross entropy loss usually used for automatic summarization. The *Contrastive Loss* was introduced to minimize factual errors. This involves having positive and negative samples during fine-tuning, and learning representations that are consistent with positive pairs (document - reference summaries) while maximizing the distance between negative pairs (document - incorrect summaries). The *Self-supervised Loss* helps the model to identify which pronouns refer to a specific speaker. It aims to determine whether two tokens belong to the same speaker. Combining these three loss, they obtained a better result for BART[Lew+20], Pegasus [Zha+20a] and T5[Raf+20] in terms of faithfulness score than previous models. Despite this, improvements need to be made as the summaries generated have factual problems.

[Hua+23] proposed SWING, an NLI-based method to address missing information in the summary while preserving dialog faithfulness. It involves using NLI to access the faithfulness of each sentence of the generated summary and whether the summary covers all reference sentences. Their proposed method is based on BART [Lew+20] pre-trained language model.

1.6 Conclusion

This chapter discusses various automatic text and dialogue summarization techniques, along with the corresponding evaluation metrics. One of the main challenges in this field is ensuring that the generated summary accurately reflects the source material. As a result, many studies focus on detecting and minimizing the number of false or misleading information in the automatic summary, known as "hallucinations."

In an industrial setting, having a summary filled with incorrect or hallucinatory information can be a major issue, rendering it useless. Due to this, we have increased our research on reducing hallucinations. Initially, we focused on eliminating hallucinations related to the entities mentioned in the text summary, which we believe is crucial for implementing this thesis in an industrial environment. With regard to automatic conversation summarization, we suggest using the dialogue structure and specific dialogue-related information to reduce the occurrence of hallucinations in the summary generated from the conversation.

This thesis makes three main contributions: (1) It analyses common errors found in abstractive summaries. (2) It reduces the risk of hallucinations related to named entities to generate summaries more faithful to the source document. (3) It reduces hallucinations in dialogue summaries by using dialogue

2 Typology of errors in text automatic summarization

Sommaire

2.1 Introduction . . . 70
2.2 Typology Definition . . . 71
2.2.1 Error typology . . . 71
2.2.2 Abstraction typology . . . 73
2.3 Evaluating summaries with the proposed typology . . . 74
2.3.1 Experimental setup . . . 74
2.3.1.1 Dataset . . . 74
2.3.1.2 Models . . . 75
2.3.1.3 Training parameters . . . 76
2.3.2 Evaluation of generated summaries . . . 76
2.3.2.1 Automatic evaluation of summaries . . . 77
2.3.2.2 Human evaluation of generated summaries . . . 77
2.3.2.3 Human evaluation of reference summaries . . . 78
2.3.3 Study of named entities in generated summaries . . . 79
2.3.3.1 Human evaluation of entities . . . 80
2.3.3.2 Automatic evaluation of entities . . . 81
2.4 Discussion . . . 82
2.5 Conclusion . . . 83

2.1 Introduction

As we discussed previously, traditional text summarization metrics such as ROUGE [Lin04], METEOR [BL05], and BERTScore [Zha+20c] do not capture inconsistencies in a generated summary with regard to the source document. [May+20] called these inconsistencies *hallucinations*. In text summarization, *hallucination* is sometimes associated to error and abstraction, which refers to a simplified or summarized representation of a concept or a set of linguistic data. This makes the understanding of the terms *hallucination* difficult. In this thesis, we consider *hallucinations* as the information not present in the source document and distinguish them from abstraction.

The research question of this thesis is: *How can we ensure that the generated summary remains faithful to the source?* To answer this question, we explore how to modify the generation process to prevent factual errors made by the system. Before jumping

to this, we focus in this chapter on how errors in automatic text summarization are categorized and which linguistic elements can be used to explicitly represent the information in a word sequence to guarantee the fidelity of the summary.

Previous works introduced different typologies to evaluate the summary generated. [PBT21] proposed seven types of errors grouped into three categories which are:

— Semantic frame errors: To capture errors located at the level of the components of a semantic frame (predicate, participant, frame elements). Predicate, Entity, and Circumstance Error are the types in this group.

— Discourse and content errors: It concerns errors such as co-references and the discourse link between two statements.

— Content verifiability errors: This group is focused on the content of the summaries. It concerns errors such as Grammatical Errors and Out of Article Errors.

Among these types of errors, some are more serious than others, especially Out of Article errors, which we call hallucination. As our work focuses on hallucination, we proposed six types of errors for generated summaries. Two are based on hallucination and non inferable information, and four for the other types of errors. By assuming that hallucination can occur if the reference summary contains information not present in the source document, we propose a typology of abstraction for the reference summary. We decided to use our typology instead of the one proposed by [PBT21] because, first, our typology distinguishes non inferable information from errors such as hallucination. Then, to the best of our knowledge, a typology of abstraction has not been explored in previous research.

We analyzed the output of different systems from the state-of-art to understand errors produced by automatic text summarization systems.

It is crucial, especially in an industrial setting, that the generated summary faithfully represent the source document. To achieve this, we decided to emphasize on identifying specific information of text summarization such as named entities which are a relatively good representation of the text. Moreover, named entities can be easily extracted from a text and provide valuable information for understanding of that text.

Therefore, we conducted a study to assess the risk of errors made by various systems in identifying these named entities.

2.2 Typology Definition

Here, we define the typologies of errors for summaries generated by automatic text summarization systems and abstraction typologies for reference summaries.

2.2.1 Error typology

Based on the assumption of [Dan05] *"A good summary is a text with non grammatical issue, non redundant with clear reference. The text must be structured and coherent."*, we proposed an error typology. This typology consists in:

— **Outside the document (HorsDoc)**: This refers to information that is not present in the source document.;

— **Agrammaticality (Agram)**: This concerns sentences that are not correctly formed or that contain grammatical mistakes;

— **Reference Error (RefE)**: This refers to instances where a proposition has the wrong subject or has been replaced by another one. It also concerns unresolved pronouns (co-reference errors);

— **Semantic contradiction (SemCtr)**: This pertains to all information that semantically contradicts the source document;

— **Non-inferable (NonInf)**: This refers to information that cannot be inferred from the document;

— **Other**: This category pertains to errors that are not included in the types cited above.

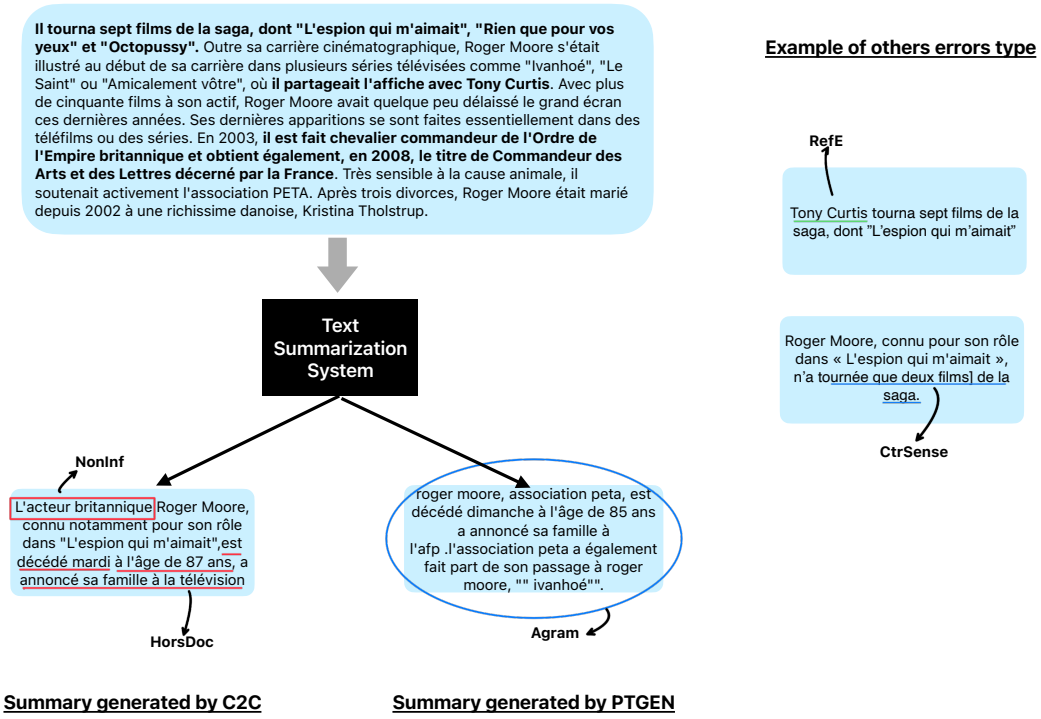


Figure 2.1 – Example of errors type in summary

Figure 2.1 shows illustrations of these different types of errors. Upon closer examination of the first summary in the image (from left to right), it was observed that it mentions a "British actor," which is not specified in the document. Although it is mentioned that the actor has been made a Knight Commander of the Order of the British Empire, it cannot be inferred that he is British. Hence, this is a non-inferrable

information present in the document. This error is classified as a NonInf error as several elements of the document are associated without being deducible. Additionally, it is stated later in the summary that the actor "died on Tuesday at 87, his family announced on television," which is not mentioned in the source document. Therefore, this is classified as a HorsDoc error. Regarding the second summary, it states that the actor only appeared in two movies in the saga, which contradicts the document's claim that he appeared in seven movies. The third summary has a RefE error because the subject has been replaced, and the last summary contains grammatical errors.

2.2.2 Abstraction typology

[May+20] showed that reference summaries of XSum dataset [NCL18] contain out-of-document information that they categorized as hallucination. Reference summaries being factual, we consider that out-of-document information in reference summaries are not errors but abstraction. Thus, we also proposed a typology for reference summaries. An abstraction refers to a simplified or summarized representation of a concept or a set of linguistic data. According to [Dye02], abstraction focuses on the most important aspects and ignores the details. It includes only the relevant information necessary to understand the system at a certain level of detail. The selection of what to include and exclude is based on knowledge and experience. Our typology categorizes the level of abstraction in reference summaries, which may contain information not in the source document. An abstraction is a grouping of information, replaced by a hypernym or inferred from the facts. We defined three types of abstraction:

— Abstractions inferable from the source document (**AbsDoc**): These are abstractions that can be inferred by reading the source document. For example, if we take the following pair document-summary:

> DOCUMENT*: Jude Bellingham is Real Madrid's new recruit. He has made an exceptional debut and is one of the players shortlisted for the Ballon d'Or.*
> SUMMARY*: Real Madrid's new recruit is in the running for the Best Player award.*
> *Real Madrid's new recruit* is an abstraction of *Jude Bellingham* and the *title of best player* is an abstraction of *the Ballon d'or*. These abstractions are inferable by reading the source document.

— Abstractions inferable from the annotator's general knowledge (**AbsInf**): These are abstractions that the annotator can infer from their knowledge and general culture. For example, we take this document-summary pair:

> DOCUMENT*: Jude Bellingham has made an exceptional debut and is one of the players shortlisted for the Ballon d'Or.*
> SUMMARY*: Real Madrid's new recruit is in the running for the Ballon d'Or.*
> *Real Madrid's new recruit* is an abstraction of *Jude Bellingham*. This abstraction cannot be inferred by looking at the source document

alone. However, if you have knowledge of soccer and, more specifically, of Jude Bellingham's path, this abstraction is inferable.

— Non-inferable abstractions (**AbsNInf**): These refer to abstractions that cannot be deduced from the document or by the annotator.

For instance, let's take the previous example. If we lack information about soccer or Jude Bellingham's path, this abstraction becomes non inferable from the source document or the reader's knowledge.

2.3 Evaluating summaries with the proposed typology

As the different typologies were introduced, we performed an evaluation of summaries generated by different state-of-the-art systems to see how many errors these summaries contain.

2.3.1 Experimental setup

Numerous studies have delved into text summarization errors in English corpora. However, there have been fewer studies conducted on French corpora. To contribute to the advancement of the community's work and determine if findings from English corpora align with those of French corpora, we chose to analyze a recently introduced corpus, Orange-Sum [KTV21]. In regards to text summarization systems, we examined three different types of systems, including RNN-based models, Transformers-based models, and Multilingual models, to see how our conclusions apply to various models. Let's now introduce the Orange-Sum dataset and the different models used in our experiment.

2.3.1.1 Dataset

For our experiment, we used Orange-Sum Abstract [KTV21] as a dataset. This was the first large-scale corpus introduced for text summarization only in French. Orange-Sum is a news corpus from the website of *"Orange Actu"* from February 2011 to September 2020. It includes five categories: France, the world, politics, automotive, and society. Orange-Sum was published in two versions:

— *Orange-Sum Title*: The authors used the news title as a summary to form the pair document-summary.

— *Orange-Sum Abstract*: It was the heading of the news that has been taken as the summary of the news. Indeed, it is the first paragraph that usually sums up the news.

Table 2.1 shows the statistics about the two versions of the Orange-Sum dataset, and Table 2.2 gives an example of each version.

Version	#doc	avg. doc. len.		avg. sum. len.		% novel n-grams		ABS	CMP
		words	sent.	words	sent.	n=1	n=2		
Title	21401	316	11	12	1.00	24.10	70.66	0.228	0.952
Abstract	30659	351	12	33	1.44	26.16	67.85	0.228	0.891

Table 2.1 – Statistics about the dataset Orange-Sum.

Orange-Sum Title
DOCUMENT: Saurez-vous retrouver qui soutient qui ? Faites le test !Le premier tour de la présidentielle aura lieu le dimanche 23 avril. Si les "grands" candidats ont vraisemblablement déjà obtenu leurs 500 parrainages, d'autres peinent encore à rassembler les signatures d'élus. Ils ont jusqu'au vendredi 17 mars à 18h00. Le second tour se tiendra dimanche 7 mai prochain. Suivront les législatives, les 11 et 18 juin.
SUMMARY: Présidentielle : quel candidat soutiennent ces personnalités ?
Orange-Sum Abstract
DOCUMENT : Daniel Cohn-Bendit est sorti de ses gonds ce mercredi dans sa rubrique sur Europe 1 et a prévenu : " Les uns et les autres, on ne se rend pas compte que l'idée d'une démocratie ouverte, une démocratie tolérante est en danger dans la situation en France aujourd'hui ". " On est pour ou contre les manifestations de la CGT, j'en n'ai rien à cirer ce matin. Ce que je veux dire, c'est que ceux qui manifestent encore après ce crime horrible, et qui disent 'tout le monde hait la police', ils sont frappadingues. Cela veut dire, en France aujourd'hui, on ne comprend pas que oui il y a des conflits, mais conflit ça veut dire dialogue, dialogue, ça veut dire compromis " a-t-il expliqué. Puis, il a mis en garde." Aujourd'hui, il n'y a plus de taquets, plus de verrous civilisationnels, et ça va dans tous les sens et je crois que c'est extrêmement dangereux ! " a lancé Cohn-Bendit ajoutant : " Je dis 'attention les enfants, vous jouez avec le feu', et dans l'histoire, le feu c'est pas, ceux qui veulent que 'Paris soulève-toi', mais ils sont dingues. Qui veut se soulever à Paris ? Ils veulent leur soulever la gueule, c'est tout ".
SUMMARY: Dans sa chronique sur Europe 1 ce mercredi, il a été particulièrement virulent contre les casseurs de la manifestation de mardi. Pour lui, " l'idée d'une démocratie ouverte, une démocratie tolérante " est en danger.

Table 2.2 – Orange-Sum Title vs Orange-Sum Abstract example

As *Orange-Sum Title* contains one-sentence summaries, we used *Orange-Sum Abstract*, which is more suitable for our task.

2.3.1.2 Models

Different state-of-the-art models based on RNN and Transformers architecture on French were used. There were state-of-the-art models at the time of experimenting.

— **PTGEN** [SLM17]: RNN-based sequence-to-sequence model with attention mechanism, pointer/generator and coverage mechanism. Details about the model were presented in section 1.2.2.1

— **CAMEMBERT2CAMEMBERT (CTC)** [Mar+20; Sci+20]: Sequence to sequence Transformer initialized from CamemBERT-base weights and pre-trained on the

automatic summarization task on the French part of MLSUM. CamemBERT French language model is based architecture on RoBERTa architecture [Liu+19b], a Transformers-based language mode, and it consists of 12 encoder layers.

— **BARTHEZ** [KTV21] : French model based on BART [Lew+20] architecture, featuring 6 bidirectional layers for Transformer-based encoder and decoder. For more details about BART architecture, refer to section 1.2.2.2.

— **MT5** [Xue+21]: multilingual variant of T5 [Raf+20], a text to text pre-trained model introduced in section 1.2.2.2. It has been initialized with the weights of mT5 trained on XLSUM [Has+21], a multilingual summarization corpus.

2.3.1.3 Training parameters

A finetuning of transformer-based models on *Orange-Sum Abstract* was done for our experiments except for BARTHEZ whose weights [1], trained by [KTV21], were used in inference. For PTGEN, we used the code provided by [SLM17] [2], which we adapted to run on French data.
In order to summarize text that contains an average of 351 words, we have set the maximum sequence length to 512 tokens for each model. Additionally, the maximum length of the output summary has been set to 128 tokens. We trained each model for 10 epochs, keeping the epoch at the end, which minimized validation loss.
At decoding step, we used a beam of size 4 as it is widely used in the literature. The decoding strategy is crucial in text generation, such as text summarization. This strategy assumes that the probability distribution of a sequence of words can be broken down into the product of conditional next-word distributions. There are various decoding methods, but we will only discuss Greedy and Beam decoding here. Greedy search selects the word with the highest probability as the next word at each step has a limitation as it may overlook high probability words that are masked by a low probability word. This can lead to a less than optimal outcome when generating text or making predictions. On the other hand, Beam search selects the m words with the highest probability and provides access to high probability hidden by low probability.

2.3.2 Evaluation of generated summaries

The purpose of the experiments is not to enhance the performance of the models but rather to analyze their outputs and assess the risk of errors when creating abstractions. Evaluating the models' performance using metrics such as ROUGE and BERTScore is still valuable. In particular, this helps us to measure how informative the generated summaries are.

1. https://huggingface.co/moussaKam/barthez-orangesum-abstract
2. https://github.com/abisee/pointer-generator

2.3.2.1 Automatic evaluation of summaries

We computed ROUGE-1, ROUGE-2, ROUGE-L, and BERTScore on the generated summaries of each system. For BERTScore, we computed two different values that correspond to BERTScore with and without the rescale parameter. Table 2.3 shows the results obtained on *Orange-Sum Abstract* corpus. The results obtained by Barthez on the *Orange-Sum Abstract* are comparable to those obtained by [KTV21]. Transformer-based systems have similar scores, while the RNN-based system, PTGEN, lags behind.

Models	**R-1**	**R-2**	**R-L**	**BERTScore**
PTGEN (RNN)	28.16	9.55	19.12	19.80 / 69.95
C2C	**31.83**	**13.22**	22.87	27.04 / 72.66
BARTHEZ	31.81	13.20	**23.07**	**28.63** / **73.25**
MT5	30.77	12.64	22.49	27.59 / 72.86

Table 2.3 – ROUGE (ROUGE-1, ROUGE-2, ROUGE-L) et BERTScore F1 for the models.

2.3.2.2 Human evaluation of generated summaries

We used the first 100 examples of the test set on which we generated summaries using the 4 models (PTGEN, C2C, BARTHEZ and MT5), i.e. 400 summaries to annotate. Considering the reference as complementary information to the source document, we followed the following annotation procedure for each generated summary:

— Read the source document associated

— Read the various summaries, highlighting any errors.

— Annotate the different highlighted errors according to the type of error.

Each annotation takes five to ten minutes, depending on the size of the source document. Table 2.4 and Figure 2.2 show the results obtained by collected human evaluation of the generated summaries.

Models	% of summaries with at least one error	# average of errors per summary
PTGEN	88	1.29
C2C	44	1.26
BARTHEZ	43	1.27
MT5	**30**	**1.12**

Table 2.4 – Error statistic of the models

PTGEN generates fewer out-of-document information (HorsDoc) than Transformer-based systems. However, PTGEN produces a larger number of erroneous summaries

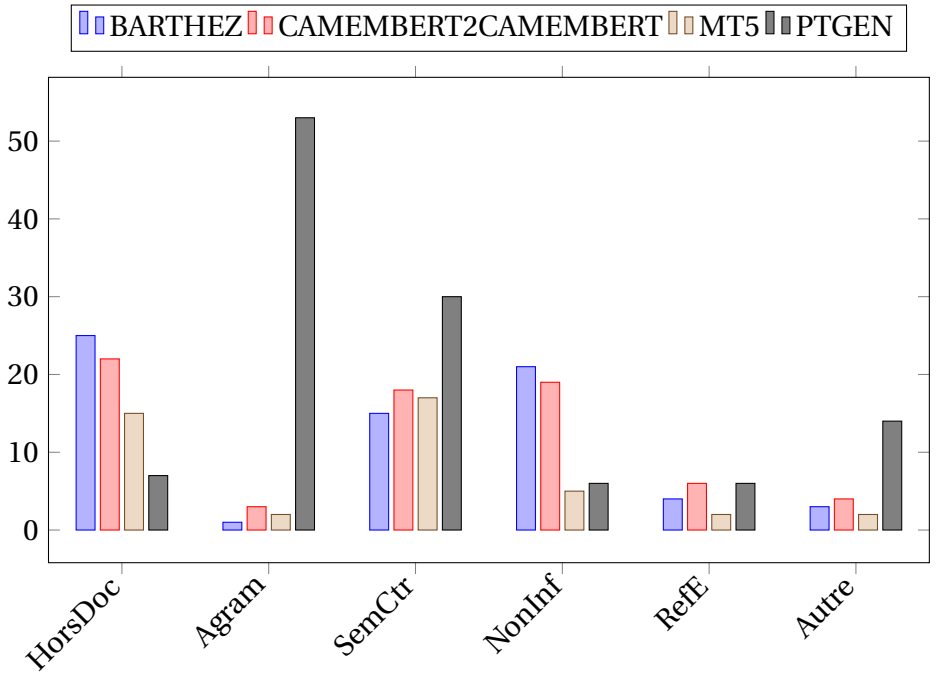


Figure 2.2 – The different types of error in each system.

due to the numerous misinterpretations (SemCtr) to which it is subject. BARTHEZ and C2C have similar frequent errors, which include information outside the source document (HorsDoc) and not directly inferable information (NonInf). This indicates their attempt at abstraction. Based on the annotation results, MT5 is the system that makes the fewest errors, as only 30% of the associated summaries contain at least one error.

2.3.2.3 Human evaluation of reference summaries

We carried out a study to evaluate the reference summaries and determine the kind of abstraction present in them. Additionally, we analyzed reference summaries to gain insight into whether errors in different systems were due to the high number of abstractions in reference summaries or not, considering the large amount of information available outside the document in various systems. To annotate reference summaries, follow these steps:

— Read the source document and its reference summary.

— Highlight any information that is not present in the source document.

— If you can infer this information from the source document, annotate it as AbsDoc.

— If you can infer it using your knowledge, annotate it as AbsInf.

— Otherwise, annotate it as AbsNInf.

Figure 2.3 shows example of annotation.

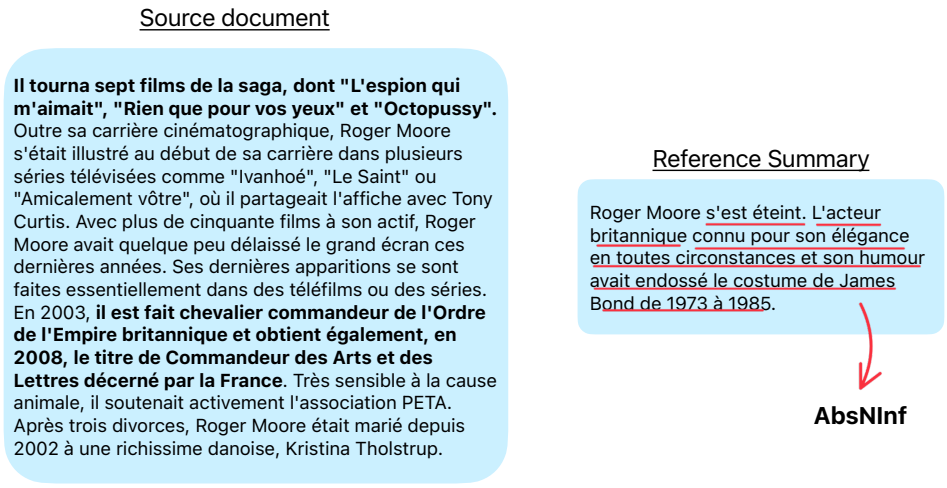


Figure 2.3 – Example of abstraction type in Orange-Sum reference summary

The annotation results are recorded in table 2.5. It's been found that 65% of reference summaries have at least one annotated abstraction, with an average of 1.17 abstractions per summary. In Table 2.5, where %abstraction represents the percentage of appearance of each type of abstraction. These findings indicate that Orange-Sum summaries often contain additional information that can't be inferred from the source document. The heuristic used to build this corpus was based on considering the caps or teasers of articles as summaries. This raises the question of its use for automatic summarization in an industrial context.

	%abstraction
AbsDoc	26.50
AbsInf	13.68
AbsNInf	59.83

Table 2.5 – Percentage appearance of different types of abstraction in the 100 annotated reference abstracts.

2.3.3 Study of named entities in generated summaries

In the previous section, we discussed automatic summarization systems that use RNN and Transformer-based architecture. However, these systems tend to generate many HorsDoc and NonInf errors, which can occur on various types of information and are difficult to study. We focused on named entities. Indeed, named entities are a

good representation of the text. In addition, much research has been done on NER, which makes them easy to extract from a text

In the following, we call an *entity* a word string belonging to a named entity categories such as a person, location, organization, and quantities.

2.3.3.1 Human evaluation of entities

We conducted a manual evaluation to identify errors involving named entities and, in particular, to identify named entities outside the document. It can be challenging to verify the accuracy of an entity that is not present in the source document without an external source. So, we considered an entity as a correct entity if it is not in the source document but appears in the reference summary. Our annotation procedure involved the following steps:

— Reading documents and summaries from different systems

— Identifying named entities such as person, location, organization, dates, quantities, etc.

— Named entities are tagged as being inside or outside the document.

— For every named entity outside the document, we examined whether it could be found in the reference summary.

An example of annotation is shown in Figure 2.4.

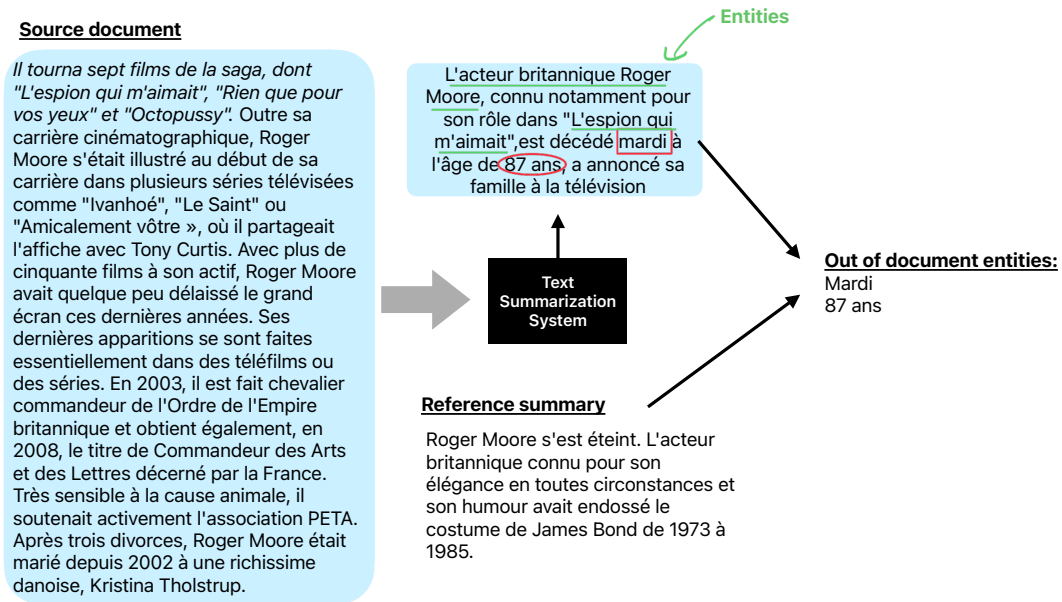


Figure 2.4 – Entities annotation example

Table 2.6 shows the results obtained by annotating the summaries generated by the 4 systems on the 100 examples used previously. In the table, $\neg Doc$ refers to

the percentage of entities outside the document, and $\neg Doc \cap \neg Ref$ refers to the percentage of entities that are not in the reference summary among the entities outside the document.

Models	$\neg Doc$	$\neg Doc \cap \neg Ref$
PTGEN	**5.4**	100
C2C	16.53	90.47
BARTHEZ	21.12	**74.07**
MT5	13.39	93.33

Table 2.6 – Percentage of OOD entities and of potentially hallucinatory entities obtained using manual evaluation

It has been observed that among the systems assessed, PTGEN has the lowest percentage (5.4%) of entities generated outside the document. However, it also has the lowest automatic score and is prone to Agram and SemCtr errors. On the other hand, BARTHEZ generates the most entities outside the source document (21.12%), of which 74% are likely to be hallucinated entities as they are not present in the reference summary. The system with the least risky output is MT5, with around 13% of entities generated outside the document. However, most of these entities (93%) are potentially hallucinated as they are not present in the source document.

2.3.3.2 Automatic evaluation of entities

We manually evaluated only 15% of Orange Sum's test data (100 examples out of 1500) and automatically considered the entities based on the same measures. Our evaluation focused on the percentage of entities outside the source document and the percentage of entities that are not in the reference. We used a NER system [ABV18] to extract entities of type PER (person name), LOC (place name), ORG (organization name), and MISC (other names). However, the system cannot recognize entities such as date, quantity, time, percentage, and numeric value. Therefore, we extracted them using regular expressions. Table 2.7 shows the results obtained.

Models	$\neg Doc$	$\neg Doc \cap \neg Ref$
PTGEN	**6.6**	97.2
C2C	9.7	87.5
BARTHEZ	15.4	**80.8**
MT5	8.6	88.7

Table 2.7 – Percentage of OOD entities and of potentially hallucinatory entities obtained using automatic approach

The automatic evaluation validates the findings obtained from the manual evaluation. The results recorded in Tables 2.7 and 2.6 are comparable. It is observed that

out of the 1500 examples, the Barthez system takes more risks than the others as it generates named entities that are neither present in the source document nor in the reference summary (approximately 15%).

We analyzed reference summaries and found that around 30% of the entities mentioned in these summaries were not present in the source document. This could be why various models predict named entities not present in the original document.

2.4 Discussion

1	***DOCUMENT:*** Triple vainqueur de la C1 sur le banc du Real Madrid (2016-2018), l'icône du football français n'a pas exclu que Neymar, Kylian Mbappé et consorts puissent enfin décrocher le Graal dimanche en finale."En tous les cas, ils sont en finale, c'est déjà un parcours exceptionnel", a estimé Zidane dans cet entretien à l'AFP accordé à Aix-en-Provence en marge de la présentation d'un programme sportif et éducatif baptisé "Zidane Five Club"."C'est la première fois qu'ils sont en finale de la Ligue des champions, Lyon est en demi-finale. C'est top pour les clubs français", s'est réjoui le Ballon d'Or 1998."On parle beaucoup de la Ligue 1 (en mal, NDLR). Je dis que la Ligue 1 est une bonne Ligue, un championnat compliqué, difficile, et là de voir les équipes françaises se débrouiller en Ligue des champions, c'est bien", a-t-il estimé.Revenant sur sa propre élimination en huitièmes avec le Real contre Manchester City (2-1, 2-1), Zidane a tout de même tiré un bilan positif de cette saison si particulière, longtemps paralysée par la pandémie de Covid-19."Finalement, d'avoir repris cette Liga, de l'avoir terminée et de l'avoir gagnée, pour moi c'était mon plus beau jour. Pratiquement le plus beau jour de ma vie", a savouré le technicien marseillais, vainqueur de son deuxième titre de champion d'Espagne comme entraîneur après celui de 2017. "Même si on a toujours envie de dire que la Ligue des champions, oui, c'est top, c'est magnifique, je n'ai pas envie de comparer, mais quand on a gagné cette Liga, compliquée, très difficile, c'était vraiment le plus jour de ma vie professionnelle. C'est très difficile à gagner", a-t-il conclu. ***SUMMARY:*** Le *PSG* a accompli un "parcours exceptionnel" en se qualifiant *mardi* pour la finale de Ligue des champions, a déclaré Zinédine Zidane à l'AFP *mercredi*, jugeant les performances de *Paris* et Lyon "top pour les clubs français" et susceptibles d'améliorer l'image de la Ligue 1.
2	***DOCUMENT:*** Il tourna sept films de la saga, dont "L'espion qui m'aimait", "Rien que pour vos yeux" et "Octopussy". Outre sa carrière cinématographique, Roger Moore s'était illustré au début de sa carrière dans plusieurs séries télévisées comme "Ivanhoé", "Le Saint" ou "Amicalement vôtre", où il partageait l'affiche avec Tony Curtis. Avec plus de cinquante films à son actif, Roger Moore avait quelque peu délaissé le grand écran ces dernières années. Ses dernières apparitions se sont faites essentiellement dans des téléfilms ou des séries. En 2003, il est fait chevalier commandeur de l'Ordre de l'Empire britannique et obtient également, en 2008, le titre de Commandeur des Arts et des Lettres décerné par la France. Très sensible à la cause animale, il soutenait activement l'association PETA. Après trois divorces, Roger Moore était marié depuis 2002 à une richissime danoise, Kristina Tholstrup. ***SUMMARY:*** *Roger Moore s'est éteint. L'acteur britannique connu pour son élégance en toutes circonstances et son humour avait endossé le costume de James Bond de 1973 à 1985.*

Table 2.8 – Examples of reference summary with out of document information.

We evaluate multiple summaries generated by different Automatic text summarization on the corpus French for text summarization corpus Orange-Sum.

Firstly, an analysis of the output of the various systems showed that, despite very interesting ROUGE scores, these systems are still affected by numerous errors linked to the factuality of the information they present.

Secondly, the results obtained suggest that the summary nature of the training data affects the models and prompts them to use information beyond the document. This aligns with the findings of [DHD20]. The results from the annotation of reference summaries highlight the importance of adapting the corpus for the automatic summarization task. The news corpora contain summaries with information that is not present in the document.

Let's take the examples in Table 2.8, which provides two examples of the Orange-Sum corpus. We can see that the different summaries contain information that is not present in their source document. Especially the second summary is clearly the headline of the article which aims to encourage the reader's interest in the article. While the document is a biography of actor Roger Moore, the reference summary talks about his death. These examples show that the reference summary of this corpus may sometimes include complementary information rather than summarize the source document. This particular scenario is typically featured in automatic text summarization datasets that are derived from news articles. Generally speaking, the headlines are deemed to be summaries. Developing a dataset specifically for automatic summarization can be a costly endeavor. As such, utilizing readily available news articles from the internet may be viewed as a viable alternative. However, considering our analysis, it remains unclear whether news corpora would truly facilitate successful summarization. Indeed, computing the percentage of entities outside the source document and those not in the reference summary amount of the OOD's entities shows that the systems reproduce the nature of the reference summaries. They have been trained to minimize the cross-entropy between the generated summaries and those of the reference. However, in the process, they generate entities that are neither in the reference nor in the source document, leading to an increased possibility of having hallucinations.

We want to acknowledge that our study has a limitation: the annotation was done by only one annotator. Therefore, it might be necessary to have additional annotation to calculate the inter-annotator agreement score. However, since our automatic and manual evaluations are consistent, we are confident that if we conduct another human evaluation, the inter-annotator agreement score will be high.

2.5 Conclusion

This chapter aims to analyze the shortcomings of generated summaries and the level of abstraction in reference summaries using two suggested typologies. Upon a detailed review of all information contained in the summary, we honed in on specific data, including named entities, which is a good representation of a text. Our findings

indicate that when automatic text summarization attempts to produce named entities not present in the source document, they are often hallucinated, as they do not exist in the reference summary. We trust that the methodology we employed in our study of named entities can be extended to other categories of information.

As we studied named entities and saw that when they are from the source document, they are most of the time hallucinated, we decided to emphasize them. Two questions were raised: How can we effectively reduce named entity hallucinations in automatic summaries? Is it possible to use the percentage of entities outside the source document as a measure to achieve this goal?

In the next section, we will explore two approaches to reduce hallucinations in text summarization, particularly on named entities. The first method consists of acting during model training, and the second, based on the percentage of OOD entities, during the generation phase.

▷ The proposal of errors typology for summary generated by automatic text summarization
▷ The proposal of abstraction typology to analyze the reference summary in terms of abstraction degree
▷ The analysis of text specific information such as named entities generated by automatic text summarization system in order to understand error on it

Table 2.9 – List of contributions made in this chapter

▷ [AFB22]: Abstraction ou hallucination ? État des lieux et évaluation du risque pour les modèles de génération de résumés automatiques de type séquence-à-séquence - **Eunice Akani**, Benoit Favre, and Frederic Bechet. 2022. In Actes de la 29e Conférence sur le Traitement Automatique des Langues Naturelles. Volume 1 : conférence principale, pages 2–11, Avignon, France. ATALA.
https://aclanthology.org/2022.jeptalnrecital-taln.1
▷ [Aka22]: Abstraction ou hallucination ? État des lieux et évaluation du risque pour les modèles de génération de résumés automatiques de type séquence-à-séquence (Résumé étendu) - **Eunice Akani**. 2022. In Journées Jointes des Groupements de Recherche Linguistique Informatique, Formelle et de Terrain (LIFT) et Traitement Automatique des Langues (TAL), pages 9-12, Marseille, France.
https://hal.science/hal-03846830v1

Table 2.10 – List of papers based on this chapter

3 Improving summary faithfulness by minimization entity hallucination risk

Sommaire

3.1 Introduction . . . 86
3.2 Named entity hallucination risk . . . 86
3.2.1 Definition . . . 87
3.2.2 Assessing the relevance of the NEHR criterion . . . 87
3.2.3 Empirical study on the NEHR criterion . . . 88
3.2.3.1 Dataset and model . . . 88
3.2.3.2 Sampling a diverse population of summaries . . . 89
3.2.3.3 Named entity extraction . . . 90
3.2.3.4 Evaluation of sampling summaries . . . 90
3.2.3.5 Human Evaluation of selected summaries based on ROUGE max . . . 92
3.2.4 Entities variation detection . . . 93
3.2.4.1 Variation detection heuristics . . . 94
3.2.4.2 Analysis of the impact of heuristics on the detection of OOD entities . . . 95
3.3 Using NEHR as a summary selection criterion . . . 97
3.3.1 Selection of summaries . . . 97
3.3.2 Evaluate the impact of summary selection based on NEHR . . . 97
3.3.2.1 Automatic Evaluation . . . 98
3.3.2.2 Human Evaluation . . . 99
3.3.3 Discussion . . . 99
3.4 Using loss function based on NEHR to ensure entities faithfulness . . . 100
3.4.1 Machine learning: training steps . . . 100
3.4.2 Proposed method . . . 101
3.4.2.1 Notation . . . 102
3.4.2.2 Loss function . . . 102
3.4.2.3 Loss function modification . . . 103
3.4.3 Impact of Loss function based on NEHR score on the summary faithfulness . . . 106
3.4.3.1 Dataset . . . 106

3.4.3.2 Model and training parameters 107
3.4.3.3 Results 108
3.4.4 Discussion 109
3.5 Conclusion and Discussion 110

3.1 Introduction

The previous chapter has demonstrated that approximately 30% of the entities present in reference summaries do not originate from the source document, potentially prompting models to fabricate information in their attempts to approximate these summaries. The phenomenon of hallucination, particularly concerning entities, has been discussed within text summarization systems based on the hypothesis that named entities are a good representation of a text. In addition, they are easily extracted from a text as many works improved the task of NER. For these reasons, we decided to study hallucination at the named entity level.

Consequently, this chapter is dedicated to investigating methods for mitigating the risk of model-generated hallucinated named entities or, ideally, preventing such occurrences altogether. The central hypothesis underpinning this section, which serves as a foundation for addressing our research inquiry, posits that reducing hallucinations related to named entities leads to a faithful summary of the source document.

To this end, this chapter introduces Named Entity Hallucination Risk (NEHR), a criterion predicated on named entities, and scrutinizes the optimal approach for its incorporation into the automated summarization task. Various methodologies for integrating NEHR into both model training objectives and subsequent phases of model implementation are explored.

In the following, we considered named entities, specific types of words or phrases in the text that refer to entities with names, including quantities, dates, numbers, and others.

3.2 Named entity hallucination risk

We propose to assess the risk of using entities and quantities incorrectly, leading to hallucinations in generated summaries. Indeed, in section 2.3.3, we showed that a selected system for text summarization generates OOD entities that are not in reference summary, so probably hallucinated.

In the following, an entity can be considered as an *in-document* entity if its corresponding word string can be found in the source document (*exact-match in-document entities)*, or if it relates to an alternative form (*e.g., New York/Big Apple*). An entity not mentioned in the source document is considered an Out-Of-Document (OOD) entity. OOD entities can be either acceptable if they can be inferred from the document (*e.g., New York → USA*) or can be considered as *hallucinations*.

In this chapter, *hallucinations* refer to the appearance of entities in generated summaries that cannot be inferred from the source document. These entities may be

mistakes or have no relevance to the context of the source document. They could also be acceptable entities when looked at through external sources but are considered doubtful by human annotators. Accepting such entities would require world knowledge beyond general comprehension, which is not directly available in the document. An entity can be incorrect in its context of occurrence in a generated summary, even if it is an in-document entity or an acceptable OOD entity.

This study aims to reduce the risk of hallucinating entities. We proposed using a criterion called *NEHR* for *Named Entity hallucination Risk* inspired by the *precision-source* metrics proposed by [Nan+21a]. We suggested an approach that involves the creation of a loss function that uses the NEHR criterion. This criterion can ensure that the model generates only those entities that are present in the given document. Additionally, the approach involves building a simple selection system that uses the NEHR criterion to identify less risky hypotheses from a large set of summaries.

3.2.1 Definition

The NEHR criterion is based on the assumption that:

> *In-document entities, especially exact-match in-document entities, are less prone to be incorrectly used in generated summaries than OOD entities.*

From this assumption, a summary-level hallucination risk, called NEHR, can be computed as the proportion of entities in that summary that are not in the source document. Given d a document and s a summary, NEHR is given as follow:

$$NEHR(d,s) = 1 - \frac{|entities \in d \wedge s|}{|entities \in s|} \tag{3.1}$$

Where named entities and quantities in d and s were detected automatically by a NER system as in [Che+21a].

Having no direct way to check if an entity considered as *risky* (*i.e.* not in the source document) is correct or not, we rely on human annotations to verify that this criterion is genuinely linked to hallucination errors on named entities in the generated summaries.

3.2.2 Assessing the relevance of the NEHR criterion

In this study, we aim to determine whether there is a correlation between our risk criterion and the faithfulness of the summary generated. Additionally, we want to investigate if there are more instances of incorrectly used entities outside the document than inside.
To achieve this, we will use the max ROUGE summary, denoted by $\hat{s_d}$, for each document d. This approach enables us to abstract from a particular system and utilize an upper bound of what current state-of-the-art systems can produce.

The following steps are involved in the empirical study designed to test the relevance of NEHR criterion:

1. Select a document/summary corpus C, train several summarization generation models on the training instances of C, generate a set S_d of alternative summaries for all documents d belonging to the test partition of C with the different models and compute the ROUGE [Lin04] and NEHR scores for all summaries $s \in S_d$.
2. For each document $d \in C$, select the maximum ROUGE hypothesis $\hat{s_d}$:

$$\hat{s_d} = \underset{s \in S_d}{\operatorname{argmax}} ROUGE(s, s_{ref})$$

3. Run a NER on each summary $e(\hat{s_d})$.
4. Manually inspect all entities e detected in summaries $\hat{s_d}$ to classify them according to 2 dimensions: inside/outside the source document; correct/incorrect use of e in $\hat{s_d}$.

The following section outlines the empirical study we conducted on the CNN/DM corpus [Her+15].

3.2.3 Empirical study on the NEHR criterion

The empirical study consists in evaluating the relevance of the NEHR criterion.

3.2.3.1 Dataset and model

We used the datasets CNN/DM and XSum datasets [Her+15; NCL18] to perform the empirical study. For the model, we used BART [Lew+20].

CNN/DM is a popular dataset for the abstractive text summarization task that comes from news articles from the CNN and the Daily Mail websites. XSum contains news articles from the BBC on different subjects. These datasets have been presented in the section 1.2.1.

BART is a transformer encoder-encoder-based model that can be fine-tuned to perform automatic text summarization. BART's pre-training consists in applying an arbitrary noising function to corrupt the text deliberately and training a model to reconstruct the original text from this corrupted version. More details about BART were presented in section 1.2.2.2.

The model was initialized with pre-trained weights from the hugging face library [Wol+20] resulting from the finetuning of BART-large on CNN/DM [1] and XSum [2]. We did not finetune ourselves to ensure we analyzed the output of the state-of-the-art systems that had already been trained.

1. https://huggingface.co/facebook/bart-large-cnn
2. https://huggingface.co/facebook/bart-large-xsum

3.2.3.2 Sampling a diverse population of summaries

The decoding method plays a crucial role in the improvement of summaries generated by systems with Transformers architecture.

Decoding strategies There are various approaches to decoding text by assuming that the probability distribution of a word sequence can be factored into the conditional distribution of the next word. For $w_1, w_2, ..., w_n$ a word's sequence, $P(w_i|w_1, w_2, ..., w_{i-1})$ and the conditional probability of the $i-$th word given the previous words in the sequence, the probability of a word sequence is given as follow:

$$P(w_1, w_2, ..., w_n) = \prod_{i=1}^{n} P(w_i|w_1, w_2, ..., w_{i-1}) \tag{3.2}$$

— *Greedy search*: The next word selected is the one with the highest probability distribution. For each time-step t, the word selected is given by:

$$w_t = \underset{w}{\operatorname{argmin}} P(w|w_1, ..., w_{t-1})$$

— *Beam search*: The *num_beams* highest probability distribution is kept at each time-step. It allows to avoid high probability hidden by low probability.

— *Sampling*: It randomly selects the next word w_t from a given set of words based on its conditional probability distribution.

$$w_t \sim P(w|w_1, ..., w_{t-1})$$

— *Temperature Sampling*: It consists in reducing the softmax's temperature to make the probability distribution $P(w|w_1, ..., w_{t-1})$ sharper. Temperature is a parameter that can be adjusted to modify the probabilities of the words generated by the model. Essentially, it affects the level of randomness in the generated text. Lower temperature tends to produce more expected tokens, whereas higher temperature yields a broader range of possibilities, some of which may be unexpected. When the temperature equals zero, the sampling becomes equal to greedy decoding.

— *Top-K Sampling* [FLD18]: It consists in taking the K most likely next words according to the probability distribution and then redistributing the probability among those K words. The value of K is chosen based on the desired balance between coherence and variability. A smaller K favors coherence, while a larger one allows more variability.

— *Top-P Sampling* [Hol+19]: It focuses on selecting the smallest set of words (or tokens) whose cumulative probability exceeds a threshold p and then redistributes the probability among them. This threshold p is a parameter that determines how much of the probability mass is considered for sampling the next word in the text generation process. Unlike Top-K, which selects a fixed number of

words, Top-p adapts the number of words it considers based on their cumulative probability, making it more dynamic.

The sampling method allows the sampling of the next word from a set of possible words to balance creativity and coherence in the generated text.

Sampling parameters For our empirical study, we generated multiple summaries using 5 different decoding strategies to select the next token from a language model. These are *Beam search*, *Greedy search*, *Temperature sampling*, *Top-K sampling* and *Top-P sampling*. We used several sampling parameters to generate a set of summaries for each document to summarize. The values of the parameters in table 3.1.

Parameters	Interval	Step	Values
Temperature	$[0.5, 1[$	0.1	$T = [0.5, 0.6, 0.7, 0.8, 0.9]$
Top-P	$[0.75, 0.96[$	0.05	$Top-p = [0.75, 0.80, 0.85, 0.90, 0.95]$
Top-K	$[40, 70[$	10	$Top-k = [40, 50, 60]$

Table 3.1 – Sampling parameters

3.2.3.3 Named entity extraction

As the NEHR criteria is based on NE, it is therefore essential to be able to extract them from documents and summaries. This can be done either manually or automatically. Given the amount of summaries we had to analyze, we used a NER system. This is FLERT [SA20; Akb+19] [3] that yields 90.93% of F1-score on OntoNotes, a large NER annotated dataset with 18 tags presented in table 3.2.

By extracting NEs from both source documents and summaries generated by the sampling method presented before, we are able to estimate the NEHR value for each summary.

3.2.3.4 Evaluation of sampling summaries

Our sampling method produces summary sets that need to be evaluated to ensure quality. The evaluation process involves using ROUGE metrics [Lin04] to determine if the summary sets contain a diverse range of summaries with varying ROUGE values. This helps us ensure that the summaries we produce are of the highest quality. To begin the evaluation process, we select the summary with the lowest ROUGE score (ROUGE min) and the highest ROUGE score (ROUGE max) from the set of generated summaries for each document-summary pair. We then compute the ROUGE and NEHR scores for both the summaries with the highest ROUGE score and the ones with the lowest ROUGE score. The results obtained have been recorded in table 3.3.

3. https://huggingface.co/flair/ner-english-ontonotes-large

List of NEs	Description	Example
CARDINAL	Cardinal number	There are **five** apples in the basket.
ORDINAL	Ordinal number	She finished in **first** place in the competition.
DATE	Absolute or relative date	We are planning the event for **April 15, 2024**.
EVENT	sport events, wars, etc.	The annual **Olympic Games** will be held in Tokyo next year.
FAC	building, airports, etc.	**Empire state building** is an iconic fac in the New York City skyline
GPE	Contries, cities, states	She visited **Paris** last summer.
LANGUAGE	-	She mastered the nuances of the **French** language during her semester abroad in Paris.
LAW	Law name	According to the **Constitution**, all citizens have certain rights.
LOC	Location non GPE	The AI lab is located in the heart of **Cambridge, Massachusetts**
MONEY	Monetary values	The company invested **$10 million** in AI R&D.
NORP	Groups	The event celebrated the diversity of **American** culture.
ORG	Companies, institutions	She works at **Google** as a software engineer
PERCENT	Percentage	The AI algorithm improved efficiency by **30%** compared to previous methods.
PERSON	People (fictional also)	**John Smith** is a renowned scientist in the field.
PRODUCT	Object, foods, etc.	The new **iPhone 15** was released yesterday
QUANTITY	Measurements	He bought **three dozen** eggs from the grocery store.
TIME	Time	The meeting is scheduled for **3:00 PM**
WORK OF ART	-	The AI-generated painting, titled "**Digital Dream**", mesmerized viewers at the gallery opening.

Table 3.2 – Description of OntoNotes NEs.

Datasets	summary set	ROUGE (R-1/R-2/R-L)	NEHR
CNN/DM	ROUGE max	57.45 / 32.59 /41.63	4.6
	ROUGE min	30.04 / 09.33 /19.47	6.0
XSum	ROUGE max	60.14 / 35.68 / 51.20	45.91
	ROUGE min	27.43 / 07.51 / 21.46	47.39

Table 3.3 – Maximal and minimal values of ROUGE on the summary set produced by our sampling method on CNN/DM [Her+15] and XSum [NCL18]. The NEHR value for the min/max summaries is also reported.

The different versions of ROUGE indicate how widely the generated summaries cover the content. However, it is important to note that significant variations in ROUGE do not necessarily mean large variations in the NEHR values for the corresponding summaries with the minimum and maximum ROUGE scores. This highlights

the fact that ROUGE may not be an accurate measure of how faithful a summary is to the original text. Actually, the ROUGE score is a measure based on n-grams overlap between the generated summary and the reference summary and does not take the source document into account during computation.

3.2.3.5 Human Evaluation of selected summaries based on ROUGE max

We manually analyzed the generated summaries to identify if our assumption that OOD entities were likely to be hallucinations was correct. We randomly selected 50 generated summaries from the test set of CNN/DM with the following constraint: each generated summary must contain at least one exact-match in-document entity and at least one that had no match (either variation of in-document entities or OOD entities). We used these constraints in order to oversample in-document and OOD entities. Using the *exact-match* constraint, we could automatically select the in-document summaries. In each summary, we manually inspected the same number of entities with exact matches to entities in the source document as those with no matches, resulting in 145 entities with exact matches and 145 with no matches.

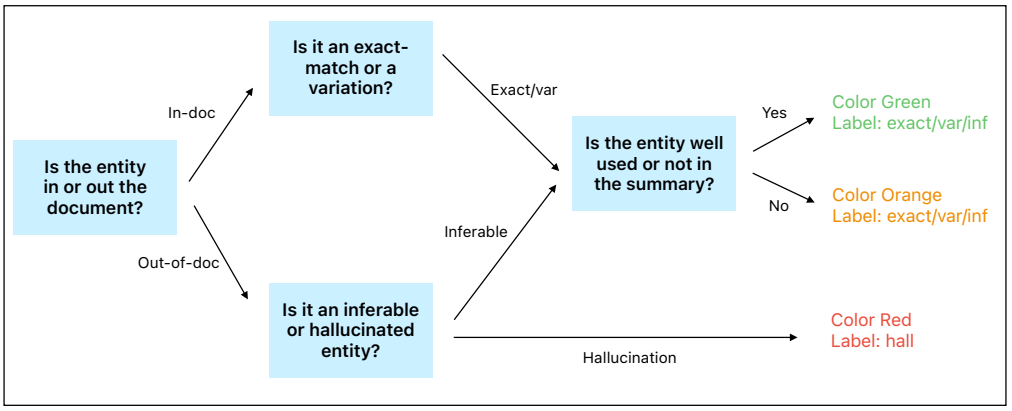


Figure 3.1 – Annotation procedure

Three annotators were asked to label each entity as correctly or incorrectly used w.r.t. the following definition: *the entity is used in the correct context according to the document*. The entities inside and outside the document were divided into two types: *exact match* or *variation* for in-document entities and *inferable* or *hallucination* for OOD entities. While *exact match* refers to entities that match exactly those in the source document, *variation* refers to entities in the document that are written with misspellings or using alternative forms. *Variation* can be a date written differently in the document and the summary. *Inferable* refers to entities outside the document whose correct use and veracity can be inferred as presented in section 3.2. We collect the annotations and report the results based on a majority vote among the annotators.

The inter-annotators agreement Cohen kappa [Coh60] of at least 0.63 was obtained for each pair of annotators. The total result is consigned in table 3.4.

A1 - A2	A1 - A3	A2 - A3
0.6833	0.6468	0.6302

Table 3.4 – Cohen kappa inter-annotator agreement [Coh60] between the 3 annotators.

According to the annotation obtained, most of the annotated entities belong to the set of in-document entities (80%); only 20% belong to OOD entities. In table 3.5, we report the percentage of correctly used entities for each subset of in-document and OOD entities. For in-document entities, 90% are labeled as "*correct*", and there are no differences between exact-match and alternative forms of entities. When dealing with OOD entities, 71% of them were considered hallucinations by our annotators. Given the dataset and systems on which we performed the experiments, this confirms our assumption that in-document entities are a good source of information for computing NEHR. Interestingly, the set of 29% inferable entities has almost the same correctness (88% *v.s.* 90%) as in-document entities. So the drop in correctness between in-document and out-of-document entities exclusively comes from hallucinations, which represent about 71% of the OOD entities. Therefore, by explicitly minimizing the risk of having out-of-document entities, we reduce the risk of hallucinations, which can lead to an increase in summary faithfulness.

Variations represent 37% of in-document entities. After careful consideration, we have decided to explore methods for identifying named entities that are variations of entities present in the document.

	Entity dist (%)	**Type**	**Type Dist (%)**	**% correct**
in-doc	79.7	*exact.*	62.87	90
		var.	37.2	90
out-doc	20.3	*inf.*	28.88	88
		hall.	71.2	-

Table 3.5 – Percentage of correctly used entities for each subset of in-document and out-of-document entities for CNN/DM dataset. Dist. is the percentage of entities belonging to each type for entities inside (*exact match, variation*) and outside (*inferable, hallucination*) the document.

3.2.4 Entities variation detection

Based on our previous manual analysis, we found that 37% of the entities mentioned in the summaries are not exact matches of the document but variations. In light of this,

we have decided to look into heuristics for detecting these variations automatically and having greater precision on which entities are outside the document to compute NEHR. Many works were proposed for this kind of task, such as entity linking, co-reference resolution [Obe19; RdP13; CGO10], but we decided to use our heuristic for finer control of variation detection.

3.2.4.1 Variation detection heuristics

We introduced several heuristics for variation detection. Here, we present them in two groups.

Text processing

— **Pre-processing**: Removes unwanted characters from the text and transforms inflections into their radical or root. In the case of an English corpus, check to see if it is not indicative of possession "*'s*" that is annoying. Also, it is used to transform inflections into their radical or root.

Example: Europeans are on the lookout for new technologies.
The entity *Europeans* will be European when applying stemming.

— **Alpha2Digit** [4]: As we consider numbers in our named entities detection, we use a Python package that converts numbers written in words into their standard form.

Example: fifty million five hundred seventy-eight thousand ⇒ *50578000*

— **SplitWord**: The idea is to process entities from a group of words. In this way, we check individually whether each word in the group belongs to the source document.

Distance-based processing

— **Spelling Corrector (SpeCorr)**: We used the tool proposed by Peter Norvig [5] that is based on the Levenshtein distance. The Levenshtein distance algorithm is a way to measure the difference between two sequences of characters. This can be useful for correcting spelling errors. The goal is to find words in a given text similar to a target word when a character deletion, replacement, transposition, or insertion is made. The tool is designed to select the most probable spelling correction for a word from a given text. As it is impossible to determine the correct spelling with absolute certainty, the tool employs probabilities. It attempts to identify the correction c from all the possible candidate corrections that maximize the likelihood of c being the expected correction, given the original word w.

$$\text{argmax}_{c \in candidate} P(c|w)$$

This tool has several limitations, notably its sensitivity to unknown words.

4. https://pypi.org/project/text2num/
5. https://norvig.com/spell-correct.html

— **BertScoreCheck** BERTScore[Zha+20c] is a method that uses contextual representations to compare entities in the source document. The goal is to identify a sentence in the source document that is similar to the entity we want to test. BERTScore is a sentence-level metric, but we have developed a technique to apply it for detecting variations and fixed the threshold at 0.8.

Example: See Figure3.2. Let's assume that the phrase $S1$ comes from the document and "Netherlands" is a variation of an entity in the document. Here, we replace it with each entity to create new phrases, then calculate the BERTScore of the resulting phrases against the starting phrase $S1$. The highest BERTScore between $S1$ and $S2$ means that Netherlands is a variation of the entity "Holland".

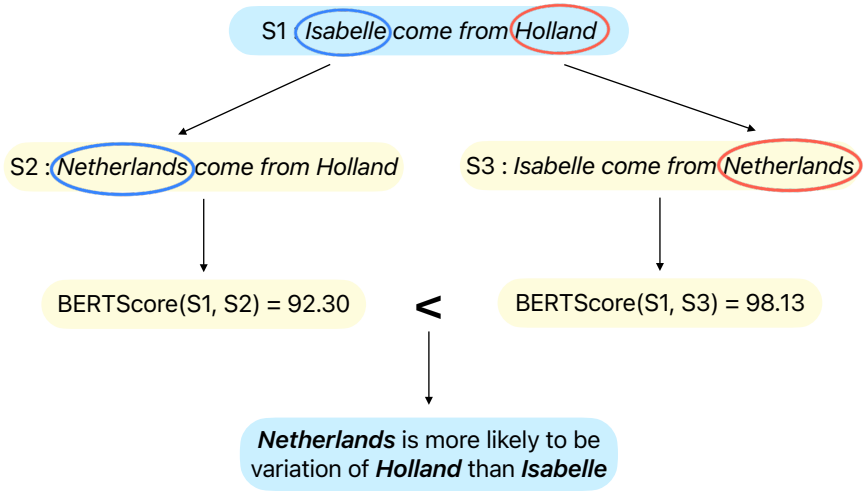


Figure 3.2 – Example of BERTScore for variation detection

When it comes to classifying named entities, heuristics play a crucial role in determining whether they belong to OOD entities or document entities. By analyzing the impact of these heuristics, we can better understand their effectiveness in accurately classifying named entities.

3.2.4.2 Analysis of the impact of heuristics on the detection of OOD entities

To create the variation detection rules, we designed a variation corpus from CNN/DM and selected 91 variations. The table 3.6 shows the percentage of variations detected by each heuristic and then by combining different heuristics. This combination is made by successively using each heuristic for variation detection in the following order: Pre-Processing + Alpha2digit > SpeCorr > SplitWord > BertScoreCheck.

Heuristics	% **of variations detected**
Pre-Processing	14
SpeCorr	24
SplitWord	30
BertScoreCheck	**39**
Pre-Processing + Alpha2digit	44
SpeCorr + SplitWord	42
BertScoreCheck + Pre-Processing + Alpha2digit	53
SpeCorr + SplitWord + Pre-Processing	**59**
All methods combined	**60**

Table 3.6 – Percentage of variations detected for each heuristic.

We then created another corpus containing 43 variations and 26 entities outside the source document to see if the heuristics detect entities as variations even though they do not belong to the source document. We found that, without using the BERT score (3.6), the percentage of entities detected was 59%, whereas with it, we were at 60%. To verify this result, we used a combination of methods to detect entities inside or outside the source document with or without using the BERTScoreCheck heuristic.

Table 3.7 shows the results. We can see that using BERTScore introduces errors in detection, as some entities outside the document have been qualified as being in the document (see the precision score on variations = 90%). As for the cost of the method, the gain obtained is not as expected. So, we only used the combination of other rule-based heuristics.

	without BertScore			**with BertScore**		
	precision	recall	f1 score	precision	recall	f1 score
Variation	**1.00**	0.60	**0.75**	**0.90**	0.65	**0.76**
Out-of-Document	0.60	**1.00**	**0.75**	0.61	**0.88**	0.72

Table 3.7 – Accuracy, Recall and F1 score for detecting variation and OOD entities

In the following, NEHR criterion will be computed using the different variation heuristics defined here.

The empirical study conducted earlier highlights the necessity of restricting the model's output to only entities present in the source document. The subsequent section outlines two approaches for achieving this.

3.3 Using NEHR as a summary selection criterion

As a reminder, our hypothesis states that using named entities solely from the source document reduces hallucinations. To achieve this, we implemented a summary selection criterion based on the NEHR score to minimize any hallucinations related to named entities. We assessed the effectiveness of this approach both through automated means and human evaluation.

3.3.1 Selection of summaries

We propose to use a two-fold criterion that considers the NEHR and model scores to select the best summary. The first step is identifying the population of potential summaries with the lowest NEHR value. The second step is to select the summary with the highest model score from that subgroup. Our approach aims to minimize the number of hallucinated entities while considering the model score. The model has been trained using the cross-entropy objective, which means that it can provide a good score to the summary it deems closest to the expected summary. By using this criterion, we can achieve a summary that is not only without hallucinated entities but also preferred by the model.

Let H be the set of summaries sampled from the model, V set of summaries with minimum risk, $P(\cdot|model)$ the probability given by the model to a summary, and $\hat{s}$ the final system output:

$$V = \left\{ x \in H | risk(x) = \min_{s' \in H} NEHR(s') \right\} \tag{3.3}$$

$$\hat{s} = \underset{s \in V}{\operatorname{argmax}} P(s|model) \tag{3.4}$$

3.3.2 Evaluate the impact of summary selection based on NEHR

We evaluate the impact of using the NEHR criterion for selecting a summary at inference time among a possible set of summaries and compare the results obtained in terms of ROUGE with three baseline selection methods:

— summary with the highest score given by the model among the summaries generated;

— the 1-best with $beam = 4$;

— the summary with the best entailment score compared to the source document as proposed in [May+20].

We also performed a human evaluation on the summary selected using the NEHR criterion and the baseline with $beam = 4$.

For this experiment, we also used CNN/DM and XSum [Her+15; NCL18] as datasets and BART-large [Lew+20] as the model. In contrast to the previous section on loss functions, we used the entire XSum test set. As in the empirical study of the nehr criterion (section 3.2.3.1), we did not finetune ourselves BART on XSum and CNN/DM and used the weight provided by the author in the hugging face library. We used the same sampling strategies with the same parameter values as presented in section (3.2.3.2) to have our set of generated summaries.

3.3.2.1 Automatic Evaluation

		R1 ↑	R-2 ↑	R-L ↑	NEHR ↓	%HallSum ↓
CNN/DM	BEAM 4	**43.74**	**20.84**	**30.44**	0.5	3.86
	BEST PROBA	41.99	18.96	28.01	2.6	20.57
	ENTAILMENT	43.61	19.69	29.26	1.62	12.92
	MIN NEHR + VAR (OUR)	42.19	19.12	28.24	**0.003**	**0.035**
XSUM	BEAM 4	**45.32**	**22.20**	**37.10**	27.67	52.48
	BEST PROBA	40.26	16.79	31.29	31.05	61.24
	ENTAILMENT	40.92	17.14	31.96	27.08	54.98
	MIN NEHR + VAR (OUR)	40.16	16.54	31.31	**6.92**	**21.49**

Table 3.8 – Evaluation on CNNDM and XSum. BEST PROBA – the summary with the best score among all generated summaries given by the model. MIN NEHR – our proposed summary selection method after using variation heuristics. %HallSum – the percentage of summary with at least one entity out of the document. ↑ higher is better, ↓ lower is better.

We use two evaluation metrics, ROUGE score [Lin04], and NEHR, to assess the quality of the generated summaries. The ROUGE score is a commonly used metric measuring the ngrams overlap between the generated and reference summaries. NEHR (see Equation 3.1) gives the hallucination risk on entities of the different selection criteria. Table 3.8 reports the results obtained using these two metrics. Additionally, we compute the percentage of summaries that contain at least one OOD entity, which can help us determine how many summaries contain entities outside the scope of the document. Our analysis reveals that the BART-Large model with beam=4 yields the best automatic evaluation scores regarding ROUGE. However, this model produces a higher NEHR value for both datasets, exceptionally for XSum [NCL18]. Our approach aims to minimize this number while maintaining almost equivalent ROUGE scores for CNN/DM. We have also observed that for XSum, there is a significant loss in ROUGE scores. However, our proposed method reduces the NEHR and the percentage of summaries containing at least one hallucinated entity. We believe that the poor performance of XSum could be because the reference summaries are not entirely faithful to the document, as 63% of XSum reference summaries contain at least one OOD entity (Table 3.9). To confirm that our selection criterion did not negatively impact

the quality of the chosen summary, we manually evaluated a subsample of the XSum test set.

Dataset	% HallDoc	#lines
Cnn/DM	29.33	11490
Xsum	63.65	11333

Table 3.9 – Percentage of reference summaries with at least one entity outside the source document

3.3.2.2 Human Evaluation

We evaluated 10 XSum test examples selected randomly. This evaluation consists in annotating the faithfulness of the different entities with respect to the document. Two annotators were asked to label each entity as true if it was used in the right context with respect to the document. An inter-annotator agreement kappa of about 0.38 was obtained. The disagreement among annotators could be attributed to the challenges they faced while annotating entities in sports-related news items where much external knowledge was required to check the correctness of an OOD entity. We held a post-annotation adjudication phase to get annotators to agree on a label for each entity. We report, in Table 3.10, the percentage of correct entities for each annotator and the adjudication. Looking at the adjudication, 61% of the entities of our method have been tagged as correct, while 48% for beam one. That means our method seems to increase the number of correctly used entities by reducing the hallucination risk.

	Annotator 1	Annotator 2	Adjudicated
BEAM 4	49.33	61.16	48.67
OUR	59.76	61.07	61.67

Table 3.10 – Average percentage of well-used entities for each system annotated. In adjudicated results, the annotators had to agree on a label for each entity.

3.3.3 Discussion

By utilizing both the NEHR and model score as a summary selection measure, we significantly reduced named entity hallucinations. Our results showed a decrease in NEHR scores while maintaining similar ROUGE scores for CNN/DM. However, when it came to XSum, there was a notable drop in the ROUGE score. This can be attributed to XSum's abstract nature and the number of new ngrams it contains. Many named entities in the reference summary did not originate from the source document. Our human evaluation confirmed our hypothesis that minimizing named entities outside

the source document using the NEHR and model score has less impact on summary quality. In fact, 60% of named entities were used in the summary compared to 48% for the summary with a Beam=4.

Our approach involves selecting summaries containing the fewest hallucinated entities after the model has already been trained. However, named entities can be hallucinated if there are no summaries in the generation space that do not contain hallucinated entities (OOD entities). As a solution, we propose an alternative approach that involves modifying the training objective to generate entities solely from the reference summary or the source document.

In the next section, we integrate the NEHR criterion into the training objective in 3 different ways and evaluate the resulting summaries.

3.4 Using loss function based on NEHR to ensure entities faithfulness

An automatic text summarization system is trained to generate summaries for a set of documents, each with an associated reference summary. The system minimizes the cross-entropy loss, which measures the dissimilarity between the candidate (generated summary) and the target (reference summary). This is the commonly used objective function to guide the learning process in text summarization.

The previous chapter discussed the challenge of hallucination, particularly on entities in text summarization systems. The reference summaries contain around 30% of entities that do not appear in the source document, which can encourage models to make up information when trying to approximate the reference summary. To address this issue, we propose a method for modifying the training objective of text summarization systems to ensure that the model only generates named entities that are present in the source document. Given that models are typically trained to minimize the discrepancy between a reference summary and the generated summary, we believe that modifying this objective to emphasize the generation of named entities holds great promise for further research.

As in previous work, we used named entities here because they are easily extracted from the text and representative. Nevertheless, the methodology can be applied to other types of information.

Before presenting our proposed approach, the following is the reminder of the different training steps to train a model.

3.4.1 Machine learning: training steps

Machine learning encompasses a range of techniques that enable automatic identification of patterns in data. These methods include supervised learning, where input data (X) is paired with corresponding labels (Y) or targets; unsupervised learning, where the system only has access to input data; and semi-supervised learning, where

some input data is labeled. Training a model involves multiple steps aimed at identifying patterns within the data and developing the ability to make predictions. Figure 3.3 illustrates the different steps involved in model training.

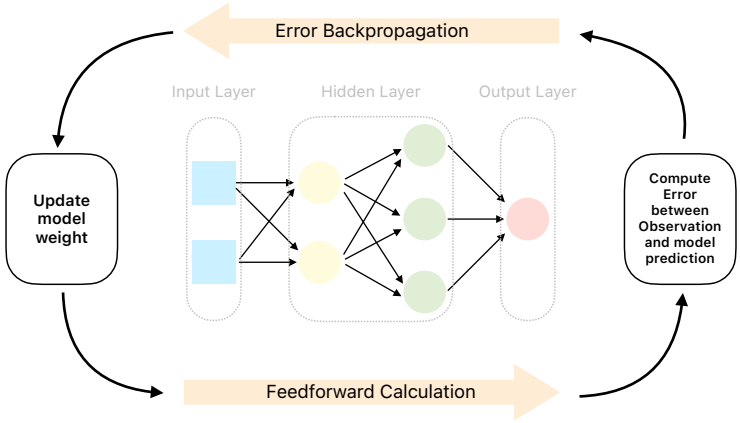


Figure 3.3 – Step to train a model

The model is first initialized at the start of learning. Once an input is presented to the network, it undergoes forward propagation through various layers via the connections between neurons until the outputs are obtained. Once the outputs are generated, a loss function compares the network's predictions with the true expected values associated with the input data. This loss function measures the distance between the network's predictions and the true values. After calculating the loss, the backpropagation algorithm propagates the output error backward through the network. It calculates the gradient of the loss function with respect to the network weights using the chain rule. The gradients are then used to update the network to minimize loss in subsequent iterations. These steps are repeated for many training examples until the model achieves a satisfactory level of performance on the training data.

When it comes to automatic text summarization, documents serve as inputs, while summaries are the desired outputs. A loss function must be minimized to achieve this, allowing the model to learn the correlation between documents and their corresponding summaries. Cross entropy is the specific loss function employed in this scenario. It measures the discrepancy between two probability distributions and evaluates how accurately the model predicts the true target values.

Now that the foundations have been laid, we present the approach.

3.4.2 Proposed method

Chapter 2 revealed that most of the system-generated OOD were not in the reference summary. This increases the risk of hallucinated entities. The previous section (section 3.2) showed that for a set of generated summaries with high ROUGE scores, there are

about 20% of their entities that are not in the source document, and only 29% of them are inferable. Therefore, minimizing the hallucinated entities is crucial.

We proposed modifying the model's training loss to encourage the system to generate only entities present in the reference summaries or in the source document. This was based on the assumption that:

OOD entities or out-of-reference summary entities are hallucinated.

In this section, we define the different loss functions to help the model generate faithful entities.

3.4.2.1 Notation

Before defining the different loss functions proposed, let's consider these notations:

— $D = \{d_1, ..., d_N\}$, the set of documents.

— $Y = \{y_1, ..., y_N\}$, the set of reference summaries associated to each document.

— $X = \{x_1, ..., x_N\}$, the set of generated summaries (hypothesis) a system generates.

— N, the number of pair examples document-summary.

— m, the maximal number of tokens for each summary ($y_1 = y_{11}...y_{1m}$).

— V the number of tokens in the vocabulary (i.e., it can be considered the number of classes).

3.4.2.2 Loss function

In Section 3.4.1, we outlined the necessary steps to train a model and discussed the use of cross entropy loss to train the text summarization model. Since our proposed loss is derived from cross entropy loss, it is essential to define the cross entropy function in a general mathematical sense.

Let's consider an example of multi-class classification. For x_i, y_i, the predicted and the reference class, respectively for example, i, the cross-entropy formula used for training the model is given below:

$$loss(x_i, y_i) = -\log\left(\frac{exp\ (x_i[y_i])}{\sum_j exp\ (x_i[j])}\right)$$

When using cross entropy as the loss function in learning, it is feasible to assign weight to a specific class. This enables the model to be sanctioned when it makes errors in these classes. This approach is especially useful in automatic classification scenarios where the dataset comprises imbalanced labels. The assigned weight serves to prioritize a minority class and offset the insufficient data that may be present.

In this case, for text classification, the cross entropy loss function will be:

$$loss(x_i, y_i) = -weight[y_i]\log\left(\frac{exp\ (x_i[y_i])}{\sum_j exp\ (x_i[j])}\right)$$

To obtain the final cross entropy loss for the model in all examples, a weighted average reduction is applied as follows: For text classification, it will be:

$$loss = \frac{\sum_{i=1}^{N} loss(i, y_i)}{\sum_{i=1}^{N} weight[y_i]}$$

These definitions can be extended to higher dimension inputs such as text summarization. We modified the loss function to have a loss for each token. For $i \in \{1...N\}$ and for $j \in \{1...m\}$, the cross entropy loss that is used to train text summarization is given as follows

$$l_{ij} = loss(x_{ij}, y_{ij}) = \log\left(\frac{exp\ (x_{ij}[y_{ij}])}{\sum_{k=1}^{V} exp\ (x_{ij}[k])}\right) \tag{3.5}$$

Now that we have defined the cross entropy loss function let's move to the proposed loss function.

3.4.2.3 Loss function modification

We introduced two new tokens ([entity] and [/entity]) to identify the start and end of named entities in the text, as our loss is based on these entities. These tokens are helpful as one entity may be divided into several tokens.

Loss depending on the named entities of the reference summary (wLoss): One intuition would be to give a large weight to the named entities of the reference in order to force the model to predict only the named entities of the reference. For this type of loss, the idea is to minimize the generation of entities that are not in the reference summary. Hence, we added weights to the named entities present in the reference summary to the computation of the cross entropy loss in order to penalize the model if it is wrong on these different entities.

Let's take W as the weight matrix for each token in the reference summary. For λ an hyper-parameter, W will be computed as follows:

$$W = (w_{ij}) \begin{cases} \lambda, \lambda > 1, & \text{if } y_{ij} \text{ is an entity or part of an entity} \\ 1 & \text{otherwise} \end{cases}$$

This equation means that for an entity, we take a weight equal to λ while the remaining tokens that are not an entity get a weight value of 1 (not a priority). The weight matrix is then added to the cross entropy loss.

$$WLM_{ij} = (w_{ij})(l_{ij}) \tag{3.6}$$

We obtained a matrix corresponding to a loss function between the generated and ground truth tokens. Then, a reduction is applied to this matrix to obtain the final loss function.

$$wLoss = \frac{\sum WLM_{ij}}{\sum(W)} \tag{3.7}$$

When evaluating a model's ability to summarize a source document, one common approach is to compare the model's summary to a reference summary that a human has written. While generating named entities from the reference summary can boost scores like ROUGE, it often fails to produce a summary that truly reflects the source document. This was evident in our previous chapter, where it was shown that the reference summary may contain named entities that were not in the source document. To sort out this, a loss function that incorporates the source document's information can compel the model to focus on the content of the source document. This promising solution can potentially enhance the model's accuracy and reliability.

Loss depending on the named entities of source document Based on the assumption that named entities of the generated summary must come from the source document, we proposed a loss function that includes the NEHR criterion as a training objective.

The idea is to guide the model during training by balancing between minimizing the error (cross entropy loss) and incorporating uncertainty (probability of having an entity that comes from the source document).

Let doc_i be the set of named entities in the source document d_i and hyp_i the set of named entities in the generated summary x_i. The probability of having generated summary entities that belong to the source document will be given by the following equation that used the NEHR equation (eq 3.1):

$$NEHR(d_i \mid x_i) = 1 - \frac{|entities \in d_i \wedge x_i|}{|entities \in x_i|} = 1 - \frac{|doc_i \wedge hyp_i|}{|hyp_i|} \tag{3.8}$$

As hyp_i can be equal to $\emptyset$, the Laplace smoothing technique was used to avoid zero division. Laplace smoothing is a statistical technique to smooth categorical data, particularly in NLP and text classification tasks. It addresses the issue of zero probabilities that may arise when estimating probabilities from limited data. Assigning a small constant value to unseen events helps avoid overly confident and unrealistic models.

Laplace smoothing adds a constant value α to all observed counts in each category to ensure non-zero probability estimates, even for unobserved categories. Let's consider the second term of the equation 3.8 as P.

$$P = \frac{|doc_i \wedge hyp_i|}{|hyp_i|}$$

Using Laplace smoothing it will become:

$$P_\alpha = \frac{|doc_i \wedge hyp_i| + \alpha}{|hyp_i| + \alpha|D|} \tag{3.9}$$

For our experiment, we take 1 as the value for the parameter α. The equation thus becomes:

$$P_1 = \frac{|doc_i \cap hyp_i| + 1}{|hyp_i| + |D|} \tag{3.10}$$

Therefore NEHR criterion will be:

$$NEHR(d_i \mid x_i) = 1 - P_1 = 1 - \frac{|doc_i \wedge hyp_i| + 1}{|hyp_i| + |D|} \tag{3.11}$$

This probability is integrated into the cross entropy loss function to encourage the model to predict only entities from the source document. Therefore, the model is penalized when the entities generated are not from the source document. Let's consider l_i the loss between the generated summary x_i and the reference summary y_i, for example i. We proposed to use a linear combination between the cross entropy loss function and the probability in 2 ways.

— **Certainty-Adjusted Loss** ($caLoss$): Here, The idea is to add, to the loss of each example, the probability of having an entity in the source document calibrated by a factor λ greater than 1 ($\lambda > 1$). The loss obtained is given as follows:

$$caLoss_i = l_i + \lambda NEHR(d_i \mid x_i) \tag{3.12}$$

Then a reduction is apply to have the final loss function:

$$caLoss = \frac{1}{N} \sum_{i=0}^{N} (l_i + \lambda NEHR(d_i \mid x_i)) \tag{3.13}$$

— **Balanced Confidence Loss** ($bcLoss$): As the values of the cross entropy loss and the values of the term $NEHR(d_i \mid x_i)$ do not have the same distribution, $NEHR(d_i \mid x_i)$ is bounded between 0 and 1. We proposed another loss based on the linear combination of these two terms to have the same value distributions on the two terms. It is given by the following:

$$bcLoss_i = (1 - \lambda) l_i + \lambda l_i NEHR(d_i \mid x_i) \tag{3.14}$$

The term $(1 - \lambda) l_i$ represents the cross entropy loss l_i weighted by $(1 - \lambda)$. This part of the loss is based solely on the model's generation ability and the actual outcome. The term $\lambda l_i NEHR(d_i \mid x_i)$ represents a penalty term. It penalizes the model based on the confidence to generate named entities that belong to the source document. If the model's confidence in its generation is low (i.e., $P(doc_i \mid hyp_i)$ is low), the penalty will be higher. Here we take λ between 0 and 1 $\lambda \in [0, 1]$. Implementing such a loss function with a λ factor between 0 and 1 significantly compromises the terms and balances the two different aspects. It makes it straightforward to understand and interpret its effect on the loss function. A value of 0 means no weight is given to the second term, essentially ignoring the term $\lambda l_i NEHR(d_i \mid x_i)$, while a value of 1 gives full weight to the

second term, focusing entirely on the term $\lambda l_i NEHR(d_i \mid x_i)$ and ignoring the cross entropy loss.

The loss of the equation x can be rewritten in another way by factoring certain terms:

$$bcLoss_i = (1-\lambda) l_i + \lambda l_i (1 - P_1) \tag{3.15}$$
$$= l_i - \lambda l_i + \lambda l_i - \lambda l_i P_{\alpha=1} \tag{3.16}$$
$$= l_i - \lambda l_i P_1 \tag{3.17}$$
$$bcLoss_i = (1 - \lambda P_1) \times l_i \tag{3.18}$$

Then we apply a reduction to obtain the final loss.

$$bcLoss = \frac{1}{N} \sum_{i=0}^{N} (1 - \lambda P_1) \times l_i \tag{3.19}$$

We consigned in Table 3.11 an overview of the different proposed loss.

Loss	Equation
$wLoss$	Eq 3.7
$caLoss$	Eq 3.13
$bcLoss$	Eq 3.19

Table 3.11 – Summary of the different loss function proposed.

A potential alternative approach that has yet to undergo testing in our experiments involves assigning greater weight to the words of an entity that is present in the source document. The intention behind this method is to discourage the model from making incorrect assumptions regarding which entities from the reference summary are actually present in the source document.

Now that we have defined the different loss functions proposed let's move on to the experiments that have been carried out.

3.4.3 Impact of Loss function based on NEHR score on the summary faithfulness

3.4.3.1 Dataset

Our experiment used the dataset XSum [NCL18], a dataset of news articles from BBC accompanied by one-sentence summaries written professionally. We chose this dataset as it is mainly used to study hallucination in text summarization. Instead of using the entire training/validation/testing set provided by the authors, we used a portion of the randomly selected training/validation/testing set. Indeed, the entire corpus requires more computation, so we decided to perform our experiment on a

subset of the dataset before applying it to a large dataset. The statistics of the dataset used are recorded in table 3.12, along with those of the original dataset.

Datasets	#doc	avg. doc. len.		avg. sum. len.	
		words	sent.	words	sent.
Original XSum	203K	392	19	21	1.00
XSum (our)	30K	295	33	17	1.00

Table 3.12 – Data distribution. Comparison between the original Sum dataset and the selected part.

As the different losses are based on entity, we must identify all entities in the source document, the reference summary, and the generated one during training. Thus, we modified the dataset to identify each entity by adding a start tag **[entity]** and an end tag **[/entity]** in the document and reference summary. Figure 3.4 shows an example of the XSum dataset with this modification.

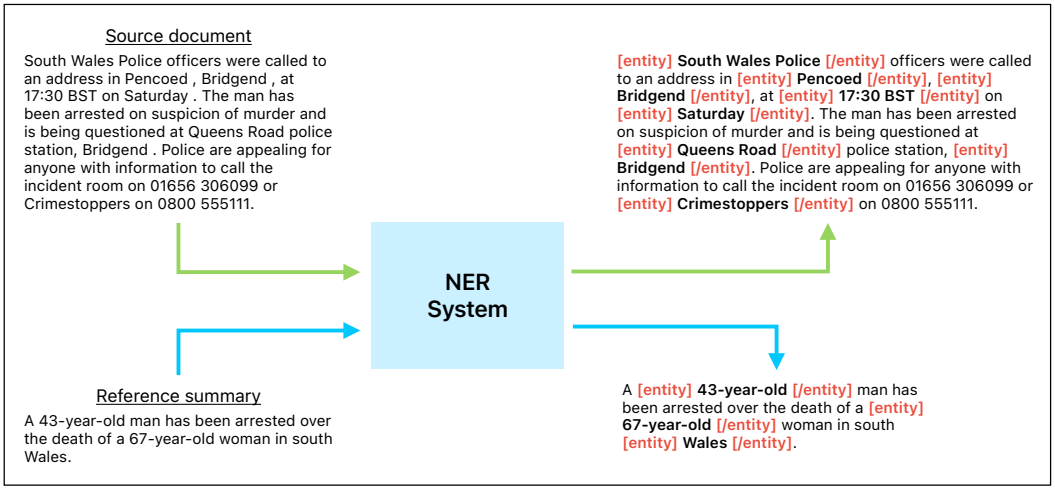


Figure 3.4 – An example of XSum with named entity tag.

To extract the entities in the reference summary and the source document we used a NER system Flert [SA20; Akb+19] [6] that yields 90.93% of F1-score on OntoNotes and presented in previous section 3.2.3.3.

3.4.3.2 Model and training parameters

To evaluate the different objective functions proposed, we used BART-base architecture [Lew+20] to train our model. BART is a denoising autoencoder transformer-based

6. https://huggingface.co/flair/ner-english-ontonotes-large

model [Vas+17] pre-trained on a large amount of data. For more information about the model architecture, refer to section 1.2.2.2. The models were initialized with the pre-trained weight provided by authors on the hugging face libraries [7].

Training parameter Each model has been trained on 10 epochs with a learning rate of 5e-5 and a batch size of 8. Only models with minimal loss on validation data were recorded and subsequently evaluated. We consider that *[entity]* and *[/entity]* each form a single token that has been added to the vocabulary for models whose data uses these tags. The named entity in the vocabulary can be a set of tokens. Thus, these tags are helpful during training to identify each token of an entity.

We trained Bart-base[Lew+20] on XSum [NCL18] dataset modified (see Figure 3.4) according to the three loss functions defined above $wLoss$ (Eq 3.7), $caLoss$ (Eq 3.13) and $bcLoss$ (Eq 3.19) by varying the λ parameter. For $wLoss$, λ must be between $[1,+\infty]$. So we trained the model for the following λ values $\{2,3,5,10\}$. For $caLoss$, λ varies between $[0,+\infty]$, so it was initialized according to the values $\{1,5\}$. And for $bcLoss$, λ must lie between $[0,1]$, so we trained the model with the following λ values $\{0.2,0.5,0.7,0.8,1\}$.
As part of our project, we trained two baseline models. The first model is trained using BART on the XSum dataset without adding entities tags. We used the basic objective function, which is cross entropy loss (eq 3.5). The second model, on the other hand, involves preserving the tags that identify the named entities in the dataset. We have also refrained from modifying the objective function in this case. We aim to establish a solid foundation for our experiments and evaluations by developing these baseline models.

3.4.3.3 Results

To evaluate the different models, we computed the ROUGE score [Lin04] between the generated and reference summaries. We also used several metrics to evaluate the named entities generated, which included entity precision, entity recall, and entity F1 score between the entities in the generated and the reference summaries. As our hypothesis was based on having entities from the source document, we additionally computed the NEHR score between the source document entities and the generated summary entities. All the scores were consigned to table 3.13.

Our results show that the ROUGE scores for training objectives caLoss and bcLoss are similar to the baseline. However, we found that ROUGE values are not necessarily correlated with variations in ROUGE scores. For wLoss, we observed that the higher the weight, the greater the loss in both ROUGE and NEHR scores. Although all training objectives had a NEHR score higher than the baseline, these scores remained well below the NEHR calculated on the reference summaries (35%).

7. https://huggingface.co/facebook/bart-base

Type Loss	λ	R1 ↑	R2 ↑	RL ↑	# ent-avg	ent-P ↑	ent-R ↑	ent-F1 ↑	NEHR ↓
Reference	-	-	-	-	3.011	-	-	-	35.83
Baseline	-	**38.75**	**15.58**	**31.14**	2.515	**0.545**	0.470	**0.505**	**21.10**
Baseline-tag	-	37.31	14.37	29.83	3.054	0.460	0.487	0.473	24.22
wLoss	2	37.27	14.30	29.73	3.058	0.461	0.494	0.477	25.25
	3	36.87	13.96	29.45	3.086	0.445	0.485	0.464	25.65
	5	36.90	13.91	29.28	3.117	0.445	0.492	0.468	26.16
	10	36.30	13.78	29.03	3.138	0.445	**0.500**	0.471	26.19
caLoss	1	37.64	14.62	30.05	3.020	0.465	0.487	0.476	24.38
	5	37.62	14.53	29.88	3.033	0.469	0.489	0.479	25.03
	100	37.82	14.65	30.03	3.024	0.473	0.489	0.481	24.72
bcLoss	0.20	38.08	14.87	30.33	3.048	0.467	0.486	0.477	23.95
	0.50	37.55	14.52	29.97	3.056	0.465	0.488	0.477	24.58
	0.70	37.61	14.66	30.04	3.049	0.462	0.485	0.474	24.42
	0.80	37.89	14.62	30.09	3.030	0.470	0.491	0.480	24.19
	1.00	37.67	14.60	29.97	3.030	0.467	0.488	0.477	24.48

Table 3.13 – Evaluation of the generated summaries. R1, R2, RL are the different ROUGE score. Entities precision (ent-P) recall (ent-R) and f1 (ent-F1) between the candidate and the reference. The average number of entities (#ent-avg) in the summary.

Cross entropy is used to evaluate the difference between two probability distributions: the distribution predicted by the model and the actual distribution of the data. In our case, the "data" are the summary tokens. Therefore, it may be expected to experience a loss in ROUGE for caLoss and bcLoss. However, since our goal is to reproduce the document entities, the NEHR scores should have been better for these two objectives, which was not the case.

On the other hand, it is reasonable to have a lower NEHR for wLoss. However, we also expected the ROUGE with the reference summary to be high, as the models were asked to reproduce the named entities of the reference. One possible explanation for the loss in ROUGE score could be the difficulty in placing the right entities in the correct locations when faced with a large list of entities to select from the source document.

3.4.4 Discussion

The obtained results were not in line with our initial expectations when we modified the model training objective to examine our hypothesis that "requiring the model to predict named entities in the document or reference would prevent errors". However, these unexpected results can be explained by the following hypotheses:

— The pre-trained models, which were trained with cross entropy as the training objective, may find it challenging to follow a different trajectory, mainly loss functions that depend on generating entities or names.

— To address this hypothesis, one option is to train the model for a longer time (over several more epochs) to incorporate the new objective better or to train the models from scratch.

Based on the information provided, it is evident that implementing changes to the model during training can be challenging due to the critical decision-making involved in selecting and defining the loss function. Our specific situation involves trained models struggling to adhere to a newly established loss function.

3.5 Conclusion and Discussion

This chapter investigated the hypothesis :

> *In-document entities, and especially exact-match in-document entities are less prone to be incorrectly used in generated summaries than OOD entities.*

In pursuit of our goal, we introduced a metric known as NEHR. Our empirical study revealed that this criterion can indicate potential hallucinations at the entity level in generated summaries. As a result, we proposed integrating this criterion into two stages of a text summarization model's design. Firstly, during the training phase, we incorporated the NEHR criterion to ensure that the model only predicts named entities from the source document. Secondly, we combined the criterion with the model's confidence score during the generation phase to refine its output. Despite our attempts to modify the cost function, we found that using the criterion downstream of generation produced more promising results.

The suitability of corpora such as XSum for automatic summarization remains a topic of debate, given the significant number of entities that are not featured in the source document. This may explain our lack of success when trying to modify the loss. In practice, the model was trained to reproduce information from the reference summary and must also avoid generating information not present in the source document. With XSum, this requirement forces the model to behave oppositely.

As further works, we want to explore loss function optimized using RL to improve the training process of generative models. The idea is to incorporate the NEHR criterion or other approaches to estimate the faithfulness of generated summary such as QA-based method. Using Reinforcement Learning from Human Feedback (RLHF) is also a path to explore. In addition, we want to explore other decoding strategies to generate words, given a model such as Monte Carlo Tree Search (MCTS) proposed by [CCK22].

▷ The proposal of a criterion based on named entities that can be used as an evaluation metric for named entities and summary selection criterion.
▷ The Study of integration of named entities based criterion during text summarization system training phase and the decoding phase.
▷ The proposal of a heuristic to detect entities variation.

Table 3.14 – List of contributions made in this chapter

▷ [Aka23]: Étude de la fidélité des entités dans les résumés par abstraction - **Eunice Akani**. 2023. In Actes de CORIA-TALN 2023. Actes des 16e Rencontres Jeunes Chercheurs en RI (RJCRI) et 25e Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL), pages 21–36, Paris, France. ATALA.
https://aclanthology.org/2023.jeptalnrecital-rjc.2
▷ [Aka+23]: Reducing named entity hallucination risk to ensure faithful summary generation - **Eunice Akani**, Benoit Favre, Frederic Bechet, and Romain Gemignani. 2023. In Proceedings of the 16th International Natural Language Generation Conference, pages 437–442, Prague, Czechia. Association for Computational Linguistics.
https://aclanthology.org/2023.inlg-main.33

Table 3.15 – List of papers based on this chapter

4 Improving faithfulness in dialogue summarization

Sommaire

4.1 Introduction . . . 113
4.2 Dialog summarization approaches . . . 114
4.3 DECODA Corpus . . . 116
4.3.1 Description . . . 116
4.3.2 Details on DECODA and statistics . . . 116
4.4 Integrating Spoken Language Understanding and Summarization . . . 120
4.4.1 Integrating semantic information to train summarization model 120
4.4.2 Task semantic information as a summary selection process . . . 122
4.4.3 Evaluation metric based on SLU . . . 124
4.4.3.1 Evaluation metric based on call type . . . 124
4.4.3.2 Evaluation metric based on named entities . . . 125
4.4.4 Experiments and results . . . 126
4.4.4.1 Dataset . . . 126
4.4.4.2 Details about the summarization and the classification models . . . 126
4.4.4.3 Evaluation Results . . . 128
4.4.5 Discussion . . . 132
4.5 Data normalisation and data augmentation for spoken conversation summarization . . . 133
4.5.1 Motivation . . . 133
4.5.2 Data normalisation and Data augmentation methods . . . 134
4.5.2.1 Automatic transcription for text summarization . . . 134
4.5.2.2 Normalize dialog summarization dataset . . . 134
4.5.2.3 Create summary using automatic system . . . 137
4.5.3 Experiments and results . . . 138
4.5.3.1 Dataset . . . 138
4.5.3.2 Models parameters . . . 139
4.5.3.3 Impact of data normalisation and augmentation . . . 139
4.5.3.4 Impact of WER on summarization performance . . . 140
4.5.4 Discussion . . . 141
4.6 SLU and data augmentation for dialog summarization . . . 142
4.6.1 Integrate call type and named entities . . . 142

4.6.2 Experiments and results 142
4.6.2.1 Experimental setup 142
4.6.2.2 call type classification results 143
4.6.2.3 Impact of task semantic information during training . . 144
4.6.2.4 Impact of task semantic information as a summary selection 145
4.6.3 Discussion 146
4.7 Conclusion 147

4.1 Introduction

In the previous chapter, we explored techniques for reducing hallucinations in text summarization and found that using NEHR as a summary selection criterion gives promising results. In this section, we decided to enhance the faithfulness in a more specific case of text summarization: conversation summarization. Conversation summarization involves generating a short, abstract version of what speakers say during a spoken interaction involving multiple parties. These conversations can take various forms, such as meetings, chats, or task-oriented dialog, which are common in customer service and technical support scenarios where conversations aim to achieve a specific goal or task. The dialogue summarization task is very important. Indeed, in an ever-changing customer service environment, where customer demands can vary rapidly, the ability to summarize dialogues enables customer service representatives to effectively manage multiple simultaneous interactions while providing accurate and relevant responses. However, dialog summarization faces challenges compared to text summarization due to differences between transcript and summary genres, errors in transcriptions, participant disfluencies, and the generative process itself, which may result in inconsistencies when generative models produce nonfactual information compared to the source.

This issue can be more widespread due to the need to rely on small language models in an industrial context to save costs and energy while keeping customer data confidential. Domain expertise can help evaluate faithfulness and be incorporated into summarization systems for improved results. Another challenge is the lack of annotated data for supervised methods. Therefore, this chapter addresses the question of How to get a faithful summary of the source document for dialog summarization, especially for task-oriented dialog summarization. We used a corpus of task-oriented dialogues recorded in an industrial setting and proposed several hypotheses:

— Integration of task-specific information can enhance the faithfulness of summary generated by models: we believe that the task information, such as call type in case of customer service conversation, can help the model to generate a faithful summary as it carries important information about the customer's need and problem-solving.

— Using automatic systems to normalize or increase data does not degrade the quality of generated summaries: To overcome the shortage of annotated data, we have opted to use LLM-based systems for generating annotated summaries. We believe that considering the performance of systems such as ChatGPT, we can achieve comparable or even better results.

— Automatic transcription can be a viable alternative to manual transcription for training summarization models: Manual transcriptions are difficult to obtain in practice due to the cost involved. Therefore, we believe automatic transcription systems are a viable alternative, particularly with improving the Automatic Speech Recognition (ASR) systems.

To verify these hypotheses, we used DECODA corpus [Bec+12], a French task-oriented dialogue dataset for dialog summarization with a Pipeline method to train dialog summarization. We selected this dataset due to its rich linguistic annotations and unparalleled authenticity, being the only dataset of this size recorded under real-life conditions to our knowledge.

We would like to highlight our significant contributions in this section, which include the following:

— Introduction of a summarization evaluation measure that uses task-specific semantic information,

— Development of a method for seamlessly integrating semantic task information into the summarization process,

— Creation of a comprehensive dataset for task-oriented dialog summarization.

Let's first explore the various approaches to dialogue summarization to better understand this thesis's contributions to automatic conversation summarization.

4.2 Dialog summarization approaches

Over time, the methods for summarizing dialogues have advanced. Nowadays, there are two main approaches for generating summaries from audio conversations. The first, known as **pipeline methods**, involves automatic transcription followed by automatic text summarization. The second, known as **end-to-end methods** [Sha+22], directly generates summaries from audio inputs without the intermediary step of transcription. Figure 4.1 provides a visual representation of these different methods.

— **Pipeline methods** To use pipeline methods, an ASR and an automatic text summarization system is essential. ASR aims to transcribe spoken language into written text. Traditional method involves several steps, starting with the extraction of audio features and the generation of phonemes with assigned probabilities. Next, a lexical module recognizes words from a pronunciation dictionary. Finally, the syntactic module generates language model constraints, and the language model estimates the likelihood of word sequences. On this transcription, text

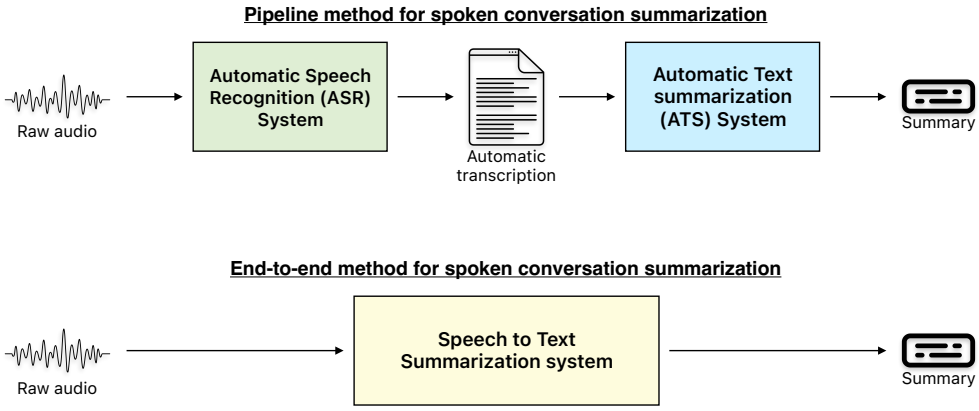


Figure 4.1 – Pipeline method vs End-to-end method

summarization model can be applied to perform dialog summarization. Thanks to deep learning with attention-based architectures and self-supervised pre-trained, the performance of both systems has been improved.
The use of ASR and an automatic text summarization system requires access to three key resources for model training, validation, and testing: audio conversations of the target domain, textual transcriptions of these conversations, and examples of text summaries based on the transcriptions. The required resources may fluctuate depending on specific system requirements, including pre-training and adaptation procedures.

— **End-to-end methods** The end-to-end approach, as highlighted in [Sha+21; Mat+23], addresses the shortcomings of the pipeline method. Unlike the pipeline method, which ignores non-verbal and acoustic cues that could enhance summarization, the end-to-end approach directly generates rich latent representations without relying on speech-to-text transcription. This approach taps into a wider range of information and avoids the limitations of pipeline systems that rely solely on automatically generated transcripts.
End-to-end methods rely on having large datasets of audio conversation and summary pairs. When such datasets are not readily available, data augmentation techniques, including speech synthesis, may be employed to generate these pairs, as suggested by [Mat+23].

Our study focuses on pipeline systems for their flexibility in using various resources and models for text or audio processing separately. Pipeline systems offer easy development and analysis without modality alignment.

Now that we have laid the foundations for our study, we can describe the dataset and the reasons behind our choice.

4.3 DECODA Corpus

Several task-oriented dialog datasets have been presented, such as AMI/ICSI [Car+05; Jan+03], DECODA [Bec+12], TODSum [Zha+21], and CSDS [Lin+21]. AMI/ICSI are both English corpora based on meeting conversations between many speakers. The main drawback of these corpora is the size. AMI consists of 137 conversations, and ICSI of 75 ones. TODSum is constructed based on MultiWOZ [Bud+18], which consists of human dialogues from seven domains. The summaries in TODSum were generated by an automated system and then corrected by humans. This dataset benefits from annotations in dialogue acts provided by Multiwoz. The drawback of this corpus is the fact that Multiwoz dialogues are unnatural dialogues created by turkeys. CSDS is a Chinese role-oriented summarization dataset designed to capture various speakers' perspectives, where summaries focus on a specific topic. DECODA is a corpus based on real-life conversations between RATP users and telephone advisors from a call center. DECODA has the special feature of being recorded in a realistic scenario with enriched annotations, including named entities and call type, compared to TODSum, which lacks realistic scenarios or industrial use cases. As a result, DECODA is the clear choice for our experiment.

4.3.1 Description

This study used the DECODA corpus [Bec+12]. This corpus contains spoken conversations from the Paris Transport Authority (RATP) call center, a company that operates part of the public transport system in Paris and its suburbs. The conversations are between several agents of the RATP customer service and users of Paris buses and metro lines. Each conversation is associated with a manual text transcription and a short summary, called a synopsis, which provides a brief overview of the main events of the conversation, including the objectives of the participants and the resolution process. Conversations cover a wide range of topics that we called *call type*, from requests for traffic information or itineraries to complaints about network operations. They may also involve requests for items lost or found on the network.

4.3.2 Details on DECODA and statistics

DECODA call type As previously discussed, DECODA offers several call types that guide the conversation and determine both the call reasons and the appropriate response from the agent. The call type provides valuable insights into the nature of the conversation and the tasks being performed. An example of call type from DECODA is *Itinerary (ITNR)*, which will correspond to a conversation around a user's route request to get from point A to point B. Another example is *Object (OBJT)*, which pertains to lost or found items on the transportation network. So, the conversation will revolve around this lost/found object. In Figure 4.2 we report the distribution of DECODA call types.

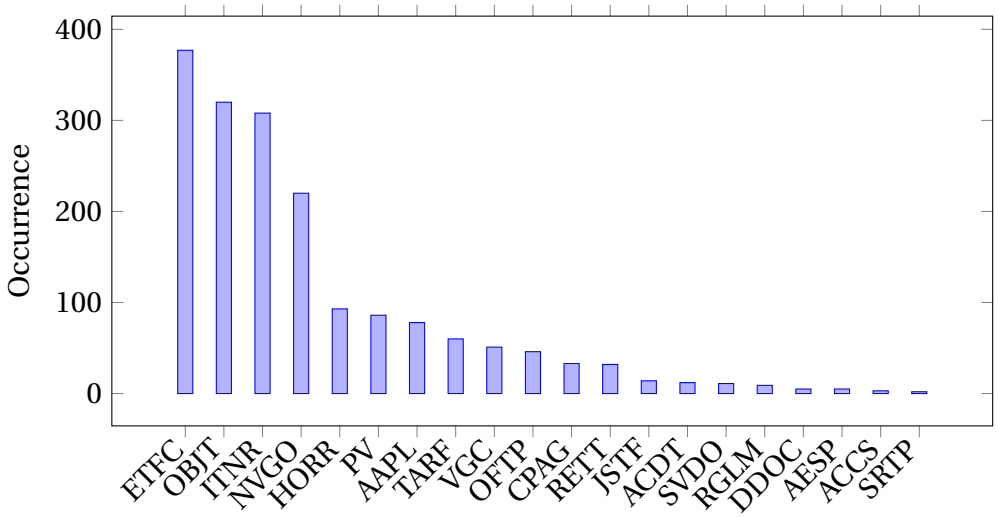


Figure 4.2 – DECODA call type distribution

Parts of DECODA The corpus DECODA consists of three parts, with annotated synopses available only in parts 1 and 3. Synopses in part 3 are longer, more detailed, and crafted in a literary style, whereas those in part 1 are synthetic and less literary. Part 2 has no summary annotation. We report the statistics and details about each part in Table 4.1.

	Synthetics Summary	Literary Summary	call type	#dialog	#turn	#conv.len	#sum.len
Part 1	✓	×	✓	1009	54.6	411.8	24.7
Part 2	×	×	✓	478	71.5	513.3	×
Part 3	×	✓	Partial	500	78.1	504.1	53.8

Table 4.1 – Description of DECODA corpus. #dialog: total number of dialogs; #turn: average number of turns; #conv.len and #sum.len average conversation and summary number of words respectively.

Table 4.2 gives an example of DECODA dataset and shows the difference between Part 1 and Part 3 synopsis in DECODA.

DECODA named entities One of the special features of the DECODA corpus is that it contains annotated named entities (NEs) belonging to a domain ontology. These include Product, Transport, and others. Table 4.3 gives the different types of NEs and an example associated. The most commonly named entity in the dataset is the address, followed by the entity corresponding to the number of the bus, metro, and other. This is quite normal, given the context of the data. In fact, given the distribution of call types, we can deduce the named entities that will be most frequently used in conversations.

DECODA Part 1 Example	**DIALOG**: *autre :* NNAAMMEE va vous répondre *conseiller :* NNAAMMEE bonjour *appelant :* bonjour monsieur *appelant :* monsieur j' ai un renseignement à vous demander *conseiller :* oui *appelant :* j' ai grand garçon handicapé mental qui a une un coupon Améthyste *conseiller :* hm *appelant :* bon évidemment ça ne dure pas un an il se démagnétise et avant j' allais euh trente rue Championnet *conseiller :* oui *appelant :* le faire changer *conseiller :* hm *appelant :* mais leur numéro de téléphone a changé auriez vous l' amabilité de lu de me donner le numéro le nouveau numéro *conseiller :* alors VGC à Championnet c' est le zéro un cinquante-huit *appelant :* oui *conseiller :* soixante-dix-sept trois fois *appelant :* soixante-dix-sept trois fois *conseiller :* hm *appelant :* très bien je vous en remercie *appelant :* monsieur *conseiller :* bonne *conseiller :* journée madame au revoir *appelant :* au revoir
	SUMMARY: numéro VGC pour faire changer carte Améthyste démagnétisée
DECODA Part 3 Example	**DIALOG**: *autre :* va vous répondre *conseiller :* bonjour *appelant :* oui allô bonjour *appelant :* je vous appelle car euh j'ai perdu mon porte-monnaie dans le... dans le train, et ça serait pour savoir quelles sont, euh comment on fait en fait ? *conseiller :* oui quel euh, quel train ? *appelant :* euh c'était le train pour aller à Montreau *conseiller :* oui il faut voir avec la SNCF madame *appelant :* d'accord *appelant :* je vous remercie *conseiller :* mais je vous en prie *appelant :* au revoir, bonne journée *conseiller :* bonne journée, au revoir
	SUMMARY: Une appelante a perdu son porte-monnaie en Gare pour aller à Montreau. Le conseiller lui rappelle donc qu'il s'agit d'une requête à adresser à la SNCF et non à la RATP.

Table 4.2 – Difference between DECODA part 1 and part 3. Example 20091112_RATP_SCD_0152 for part 1 and 20110704_RATP_SCD_0018 for part 3 from DECODA.

Others DECODA annotation The corpus has named entity annotations, morpho-syntactic labels, lemma annotations, and syntactic dependencies. Moreover, the DECODA corpus was anonymized to guarantee confidentiality. Table 4.4 gives an example of DECDOA conversation annotation. In the example, we can see *NNAAMMEE* that corresponds to the name of a speaker that has been anonymized.

All these annotations make DECODA a rich dataset for our study of task-oriented dialogue summarization.

List of NEs	Description	Example
A	Adress	Avez-vous un bus qui passe à la **rue Ledru-Rollin** ?
CP	Zip code	Mon code postal est le **92500**
D	Date	Votre pass est valable du **1er au 30 novembre**
H	Time	Le bus passe à **10h10**
HU	Human (Person)	Bonjour **NNAAMMEE**
L	Location	J'habite dans le **Var**
M	Monument	Je pars à l'**église de Laye**
ORG	Organisation	La gare de **SCNF**
P	Price	Le ticket coûte **2 euros**
PRD	Product	C'est un ticket **Améthyste**
T	Transport	Prenez le bus no **128**
TEL	Telephone	Appelez plutôt le **08 00 00 00 00**

Table 4.3 – Description of DECODA NEs

Turn id	Word	Disfluency	POS	NE	dep	lemma
1	NNAAMMEE	-	NP	B_HU	ROOT	ANONYM
1	bonjour	-	ADV	-	NOLINK	bonjour
2	NNAAMMEE	-	NP	B_HU	ROOT	ANONYM
2	bonjour	-	ADV	-	NOLINK	bonjour
3	allo	-	ADV	-	NOLINK	allo
4	pour	-	PREP	-	ROOT	pour
4	aller	-	VINF	-	OBJ	aller
4	sur	-	PREP	-	P_OBJ_LOC	sur
4	Dammartin-Juilly-Saint-Mard	-	NC	B_A	OBJ	Dammartin-Juilly-Saint-Mard
4	hein	DISF_M	PRES	-	NOLINK	hein
4	c'	-	CLN	-	SUJ_IMP	ce
4	est	-	V	-	ROOT	être
4	le	-	DET	-	DET	le
4	nom	-	NC	-	ATS	nom
4	de	-	PREP	-	DEP	de
4	la	-	DET	-	DET	le
4	gare	-	NC	-	OBJ	gare
4	où	-	PROREL	-	P_OBJ_LOC	où
4	vous	-	CLN	-	SUJ	CLN
4	allez	-	V	-	ROOT	aller
4	c'	-	CLN	-	SUJ_IMP	ce
4	est	-	V	-	ROOT	être
4	ça	-	PRO	-	ATS	ça

Table 4.4 – Example of manual annotation of the DECODA corpus

4.4 Integrating Spoken Language Understanding and Summarization

This section presents two ways to integrate semantic task-specific information into the summarization process. It also introduces a new criterion based on semantic task information.

Given our focus on goal-oriented human-human conversations, we can leverage the same semantic information that has been proposed for performing Spoken Language Understanding (SLU) in human-machine dialogue systems for tasks like transportation reservation (e.g., ATIS corpus [HGD90]) or restaurants/hotels (e.g., MEDIA corpus [Dev+04]). Typically, three semantic levels are defined in this type of study [Lee+18]:

— **domain**: the semantic context of the dialogue, such as flight bookings in the ATIS corpus. Our study pertains to public transportation in Paris for the DECODA corpus.

— **intent**: the nature of the request made in the dialogue, like confirmation or information inquiry. In the DECODA corpus, intents are assigned to the entire conversation, representing *call types* such as *itinerary requests* or *lost item claims*.

— **entity/value pairs**: semantic relationships within intents, such as a destination in an itinerary request. In the DECODA corpus, entities include *location*, *bus numbers*, *time*, and s*ervice ID numbers*, among others.

We believe that by incorporating task-specific semantic information into the summarization process, we can effectively address inconsistencies related to this information in the data. Given that our application corpus only has one domain, we will focus on the call type and entity level in this section. Using call type and named entities is an excellent choice for our study, as the DECODA corpus includes annotations for both. Additionally, leveraging named entities provides a practical framework for our methodology. By using entities, we can avoid the challenge of determining the correct representation of information and can instead rely on existing methods for the NER task. Nevertheless, other semantic information, such as semantic frames, can be used in our methodology.

We will introduce two methods for performing SLU/summarization: one during the training phase and one after, which can be readily applied in real-world scenarios.

4.4.1 Integrating semantic information to train summarization model

Identifying the different call types and corresponding concept labels is often necessary to analyze and understand the content of conversation recordings or transcripts. However, this information is not directly available and needs to be inferred. This can be achieved in a few ways, including implementing a separate prediction system prior

to generating the transcript, integrating the call types and concept labels into the input data, or using a multitask approach during summarization to identify and label the different call types.
Let D be the input dialog, S the generated summary, E the set of entities, and C the call type. We consider the following methods:

1. **Baseline**: No explicit semantic information is used to generate the summary.
2. **Pipeline**$_X$: Predict semantic information X from dialog D such as $X = \text{intent}(D)$, then condition summary generation on X

$$S = \text{summary}(D, \text{semantic-information}(D))$$

The semantic information can be call type, named entities or other type of semantic information. In our case we used the call type C and predict them from the dialog $C = \text{intent}(D)$:

$$S = \text{summary}(D, \text{call-type}(D))$$

3. **Multitask**$_{C,E}$: Generate both semantic labels such as call type or named entities and summary directly from conversation transcript. This result in two systems.

$$\{C, E\}, S = \text{summary} \circ \text{semantics}(D)$$

Figure 4.3 gives more inside into the integration of the task semantic information. In our experiments, a language model was fine-tuned on an automatic summarization task corresponding to each scenario.

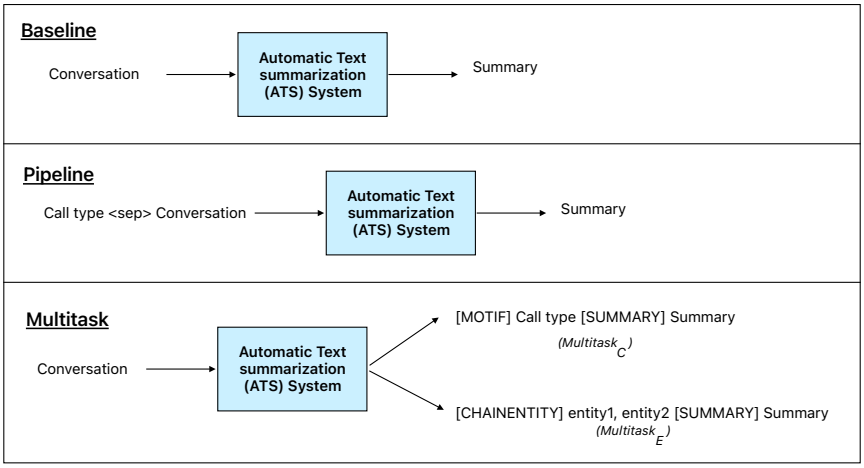


Figure 4.3 – Integration of semantic information about the task during summary generation process

[Nar+21] proposed to generate named entities before generating the summary and

apply their technique on text summarization to reduce hallucination. Following their approach, we decided to apply the same technique to dialog summarization. While [LC21] recommended using only named entities that identify people to model the dialogue and its participants, we opted to incorporate all types of named entities, as suggested by [Nar+21] for text summarization. This decision was made because we believe that other named entities are crucial in solving task-oriented dialogues, such as those found in customer service conversations.

Here, we integrate call types and named entities in the model's training. We believe that any semantic information can be integrated into the summarization process in order to have a good semantic representation of the conversation regarding the information.

As demonstrated in chapter 3, using the NEHR criterion for summary selection yielded a summary that remained faithful to the source document regarding named entities. Regarding tasks-oriented dialog summarization, incorporating additional factors such as call type as a selection criterion may offer a viable option for enhancing the fidelity of the summary. Our upcoming section will delve into this approach further.

4.4.2 Task semantic information as a summary selection process

Alongside incorporating semantic information directly into the training phase, we suggest using it post-generation to choose the most semantically reliable summary per our models. Multiple outputs with different characteristics can be obtained by tweaking parameters, like temperature in text generation, as in chapter 3 section 3.2.3.2. We propose a selection method based on call type prediction and the risk of hallucination on task-related entities. The idea is to select, for each example, the best summary regarding each criterion among a set of generated summaries.

Selection based on call type Based on the hypothesis that, generated summary must have the same call type than the transcription of the conversation, we suggest using call type as a summary selection criterion. Our method involves using a text classifier to predict the call type of the generated summary and comparing it to the predicted call type for the entire conversation transcript. The hypothesis is therefore as follows:

> *if the predicted call type for the summary is consistent with that of the full transcript, then the summary is likely to be semantically coherent with respect to call type.*

We handle multiple call types and uncertainty in call type classification by avoiding binary comparison and, instead, calculating the divergence between the probability distribution on all call types for both summary and dialog classifiers. We implement the Kullback-Leibler (KL) divergence [KL51] between the probability distributions

of call types. The KL divergence is a statistical distance measure that quantifies the dissimilarity between two probability distributions. It evaluates the difference between a probability distribution and a reference distribution. The KL divergence is one of the most commonly used measurements. However, there are several other alternatives for measuring similarity between distributions, such as Divergence de Jensen-Shannon, which is a symmetrical measure of the divergence between two distributions. This method is useful when symmetry is crucial, and there is a need for a symmetrical divergence measurement. We opted for the KL divergence because it emphasizes the relative difference between two probability distributions.
Let $G = \{g_1, ..., g_n\}$ be the probability distribution given by the call type classifier on the generated summary, $R = \{r_1, ..., r_n\}$ the one obtained by the classifier on the entire conversation for n the number of call types and CT the set of call types, the KL divergence between G and R is defined as follows:

$$D_{\mathrm{KL}}(G \parallel R) = \sum_{x \in CT} G(x) \log\left(\frac{G(x)}{R(x)}\right) \tag{4.1}$$

It is now possible to select the summary that minimizes the D_{KL} distance among the set of generated summaries.

Selection based on NEHR NEHR criterion as a selection criterion allows to have a summary for faithful in terms of NEs for text summarization. Then, we suggest using it for dialog summarization. As a reminder, NEHR measures the proportion of OOD entities and it was defined on Equation 3.1.
When a summary generation system produces a named entity not in the original document, it increases the risk of a model's over-generation error. After observing that selecting the summary with the least risk of hallucinated named entities produced favorable named entity evaluation results (Section 3.3), we decided to investigate the same experiment on dialogue summarization.

Combining D_{KL} and NEHR as a selection criterion The purpose of this study is to have a summary of faithful regard to the semantic information. Thus, we hypothesize that using a criterion that combines multiple criteria will result in a summary consistent with the source document. As we used the named entities and the call type as the semantic information we want to evaluate, we hypothesize that combining NEHR and D_{KL} as a selection criterion may result in a summary consistent with the source document in terms of call type and NEs.

Let H be the set of summaries sampled from the model, V set of summaries with the minimum NEHR, m the minimum NEHR value, $D_{\mathrm{KL}}(x \parallel R)$ the KL divergence between a summary s associated to the conversation transcription R, and $\hat{s}$ the final system output:

$$m = \min_{x \in H} NEHR(x) \tag{4.2}$$

$$V = \{x \in H | NEHR(x) = m\} \tag{4.3}$$

$$\hat{s} = \min_{x \in V} D_{\mathrm{KL}}(x \parallel R) \tag{4.4}$$

We select summaries with the minimum NEHR scores between the generated summary and the document to determine the optimal summary. From this selection, we then choose the summary with the lowest D_{KL} value with the document. While we initially considered a linear combination of these criteria, it was not possible due to their differing value ranges. Specifically, D_{KL} varies between 0 and $+\infty$, while NEHR falls between 0 and 1.

Our proposal involves the integration of task semantic information, including call type and named entities, in order to improve the coherence of text summaries regarding this information. To ensure that this approach produces the desired results, we suggested four semantic metrics that can be used to evaluate the generated summaries. These metrics are outlined below.

4.4.3 Evaluation metric based on SLU

Assessing the evaluation of automatic text summarization is a complex task, as we have emphasized throughout this document. Traditional metrics such as ROUGE [Lin04] or BERTScore [Zha+20c] do not account for information that deviates from the source document. While ROUGE computes n-gram overlap between the generated and reference summaries, BERTScore compares the two and assigns a score based on their similarity. These measures are sensitive to paraphrasing and the presence of new words, which are essential factors in abstract summarization. Therefore, additional metrics are necessary to evaluate the semantic aspects of summarization that lead to consistent results. In our study, we focus on task-related semantic information, such as call types and named entities, and propose specific metrics to evaluate each aspect.

4.4.3.1 Evaluation metric based on call type

Based on the assumption that a generated summary must have the same call type as the reference call type, we propose the **CT-Acc** metric. The **CT-Acc** metric measures the accuracy of the call type classifier used on the generated summary compared to the reference call type. We assume that the summary is more coherent with the call type classifier if the accuracy is higher. Let takes the examples on Figure 4.4.

We passed for generated summaries in a call type classifier to obtain the associated call type and then compared them to the reference call type. Here, one mistake have been done: square call type were replaced by pentagon call type. Then, by calculating the accuracy, we obtain the **CT-Acc** of 0.75.

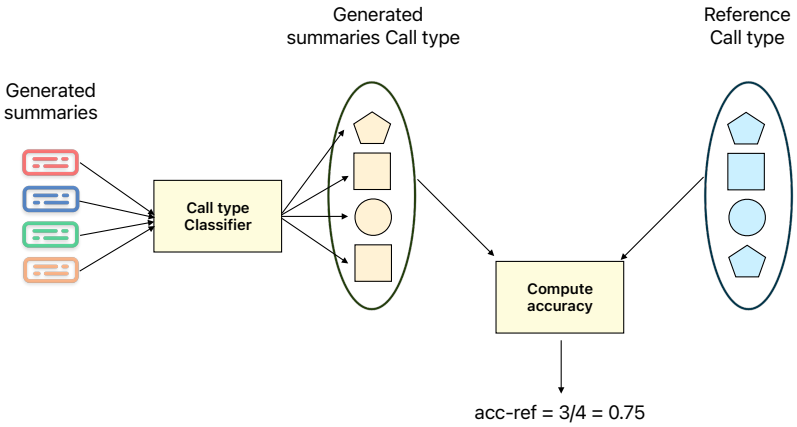


Figure 4.4 – Step to evaluate generated summary in term of CT-Acc

4.4.3.2 Evaluation metric based on named entities

We proposed to use a metric based on named entities to assess the generated summaries compared to the reference summaries. We aim to ensure that the generated and reference summaries contain the same entities. To accomplish this, prec_t, recall_t, and F1_t were introduced by [Nan+21a] to measure the entity-level accuracy of the generated summary. We have since introduced NE-P, NE-R, and NE-F1 metrics, which represent the precision, recall, and F1 between entities in the generated summary and those in the reference summary. We believe that higher values for all these metrics indicate a more coherent summary. The difference between our metric and theirs is in what is considered a named entity. They exclude certain types of entities, such as dates, times, and numerals, due to large variations in representation and difficulty in determining a match. However, we keep them as they can be crucial in some scenarios. For instance, the DECODA corpus, which is based on conversations between users and the Paris transport network, contains route requests based on bus numbers or dates. Ignoring these entities can result in semantic errors. To identify variations on quantities or named entities, we adapted our entity-level variation detection approach from chapter 3, section 3.2.4 for our needs. Our named entities metric were based on the CamemBERT [Mar+20] language model.

For G_e the set of entities in the generated summary and R_e the set of entities in the ground truth summary, NE-P, NE-R, and NE-F1 are computed as follows:

$$\text{NE-P} = \frac{|G_e \cap R_e|}{|G_e|} \quad \text{NE-R} = \frac{|G_e \cap R_e|}{|G_e|} \quad \text{NE-F1} = 2 \times \frac{\text{NE-P} \cdot \text{NE-R}}{\text{NE-P} + \text{NE-R}} \tag{4.5}$$

Our proposed metric can help measure the performance of the summarization selection. Indeed, minimizing NEHR is expected to reduce the recall NE-R and increase precision NE-P.

In the following, we will use these four measures alongside ROUGE and BERTScore to evaluate the summaries we generated.

4.4.4 Experiments and results

4.4.4.1 Dataset

Given the unavailability of a synopsis for part 2 of DECODA, we proceeded with our experiments using examples exclusively from parts 1 and 3. This approach was chosen to ensure the relevance and validity of our experiment. We focused on instances from parts 1 and 3 that were annotated with call types. We divided the dataset into three subsets: Train, Val., and Test. The Train subset was used for model training, Val. for development, and Test for evaluation purposes. We employed stratification based on call types during the partitioning process to guarantee that each call type was evenly represented across all partitions. For additional insights, please consult Table 4.5, which provides our computed statistics on the dataset.

Stats	Train	Val.	Test
# examples	717	99	140
# conv. words	486.3	487.9	465.4
# syn. words	28.8	30.3	29.5
# conv. tokens	608.2	604.7	579.8
# syn. tokens	35.9	37.5	36.2

Table 4.5 – DECODA dataset distribution. #examples is the number of examples for each partition. #conv. words and #syn. words refer to the average number of words in the conversation and the synopsis respectively. #conv. tokens and #syn. tokens refer to the average number of tokens for the conversation and the synopsys respectively.

For research purposes, the different splits of the dataset will be available further.

4.4.4.2 Details about the summarization and the classification models

Here we describe the dataset and the model used for our study.

Automatic summarization As the dataset selected for our studies was in French, and given the results obtained by the transformer-based models, we had two options for choosing the model: either a pre-trained French model or a multilingual model. We decided to use a pre-trained French model as a multilingual model, which can be biased by the amount of data in each language.

Regarding the state-of-the-art, we employed BARThez [KTV21]. This sequence-to-sequence model has been pre-trained on various French corpora, including CommonCrawl, Wikipedia, and NewsCrawl, among others corpora. BARThez was introduced for automatic text summarization tasks. This makes it a good choice for our study.

BARThez is a Transformer-based model with a similar architecture to BART [Lew+20]. It features six Transformers encoders and six Transformers decoders in its BARThez-base version. We used BARThez-base with the pre-trained weights provided by the model's creators [KTV21], which can be found in the Hugging Face Transformers library. [1]

In Table 4.5, we consigned the average number of tokens for both conversations and synopses, which were computed using the BARThez tokenizer. Each model for text summarization was trained during 12 epochs using AdamW optimizer with a learning rate of 5×10^{-5}. The maximum input length (corresponding to the conversation) and output length (corresponding to the synopses) were set to 1024 and 128, respectively.

To train **Multitask**$_C$, we added the call type to each reference synopsis by concatenating the call type to the synopsis. Moreover, for **Multitask**$_E$, it was the named entities that have been added. This modification allowed us to adapt the data to the desired model, as shown in Figure 4.3.

For **Pipeline**$_C$, we needed to associate the call type with each conversation. We used a call type classifier trained with k-fold cross-validation to achieve this. We trained the classifier using 75% of the training set and made predictions using the remaining 25%. We repeated this process four times to obtain all the call types used to train the summarization model (see Figure 4.5 for the process). For the text summarization model, the call type and the conversation have been separated by a special token to help the model focus on the call type when generating the candidate summary, as shown in the example in Figure 4.3.

Call type classification The integration of call type to text summarization requires a call type classifier. Indeed, the call type classifier is used to evaluate the generated summary using the KL divergence and also for the **Pipeline**$_C$. We used CamemBERT [Mar+20], a French transformer-based language model pre-trained on large data, to train our call type classifier. Based on the architecture of RoBERTa [Liu+19c], an English pre-trained language model, CamemBERT has two versions that differ in the number of Transformers Encoder layers. We trained in the base version to predict call type. We created two classifiers leveraging CamemBERT. The first classifier is designed to identify call type from input conversation **Conv-CT-classifier**, while the second classifier identifies call type from synopsis **Syn-CT-classifier**. The model was trained using an input length of 512 and 128 for conversation and synopsis, respectively. We opted for 512 instead of 1024 for conversation as CamemBERT's architecture only supports 512 tokens as input. Both classifiers were trained for 13 epochs, with a batch size of 8, and we saved the checkpoint with the lowest loss function during the validation phase.

Domain-Specific Named Entity Recognition To integrate named entities in **Multitask**$_E$, perform summary selection using NEHR and evaluate NEs, we need to extracted named entities. As we said, DECODA has 12 named entities specific to the

1. https://huggingface.co/moussaKam/barthez

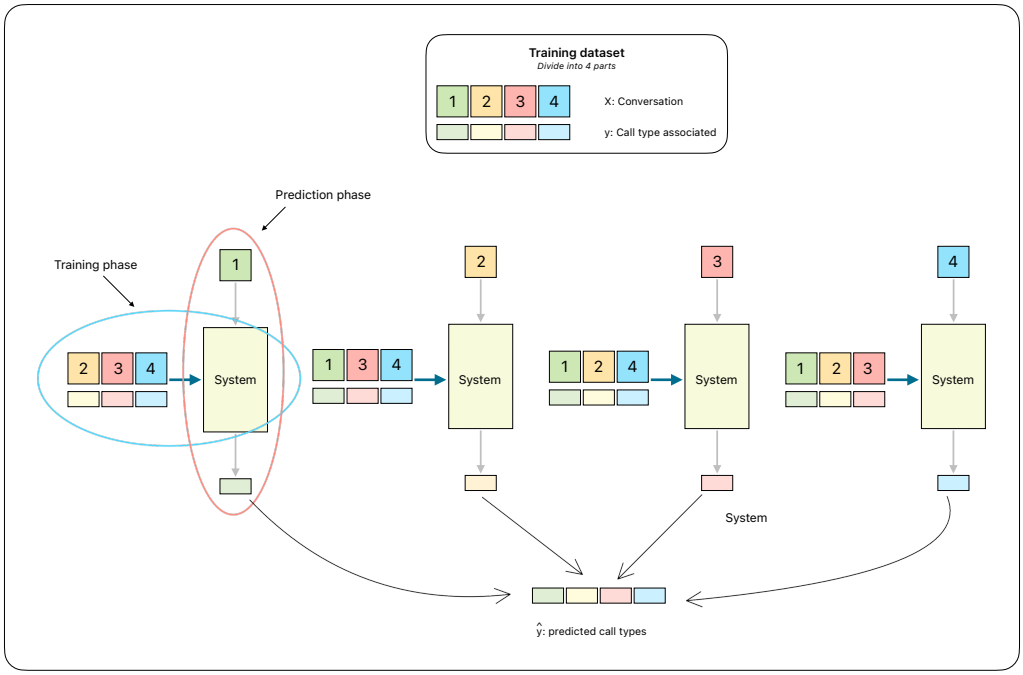


Figure 4.5 – Prediction of call type on the training dataset.

domain. To accurately recognize and classify these entities, we trained CamemBERT [Mar+20] for Token classification to perform named entity recognition. We choose CamemBERT as it gives good results for the task. Our NER dataset was meticulously designed using the conversations. Each conversation was divided into speech turns with associated named entities. In this way, each turn represents an example of our dataset for training the NER model. For the model training, we take the maximum length that CamemBERT can have as input, 512. We then trained the NER model for 14 epochs with a batch-size of 14, using a default hugging face learning rate of 5e-5 for the optimizer AdamW. After conducting evaluations using Seqeval [Nak18], the model achieved a micro F1 of 0.93 and a macro F1 of 0.84 on named entity recognition in conversations against the gold standard.

4.4.4.3 Evaluation Results

Call type classification After training the two classification models **Conv-CT-classifier** and **Syn-CT-classifier** for call type classification, we have compiled the results and presented them in Table 4.6. Due to the unbalanced nature of the data, we used the accuracy (Acc.) and F1 score (M-F1, W-F1) to evaluate each model. Table 4.6 shows that the conversation-based classification model outperformed the synopsis-based model. This can be attributed to the additional information that the conversation-based model was able to access, which provided useful insights into the purpose of each call. As synopses of DECODA part 1 are shorter, they may not contain

enough information to infer the call type. Furthermore, it is worth noting that the multitask model **Multitask**$_C$, which predicts entities simultaneously with the synopsis, yielded notably weaker results than the classifiers. This may be due to the complexity of the task the model have to perform. Actually, the model have to generate the call type before the generated summary.

System	Acc.	W-F1
Conv-CT-classifier	86	86
Syn-CT-classifier	83	81
Multitask$_C$	79	76

Table 4.6 – Call type classification results using the conversations as input Conv-CT-classifier, the synopses as input Syn-CT-classifier and the multitask model Multitask$_C$. Acc. is the accuracy and W-F1 is the weighted average F1.

Automatic summary based on call types In section 4.4.1, we elaborate on the various approaches for integrating the call type into the summarization system. To generate the call type, we include it in every reference summary for Multitask$_C$, followed by the synopsis. For Multitask$_C$, we extracted NEs from the reference summary and integrate them to the every reference summary (refer to Figure 4.3). In the case of Pipeline$_C$, the model receives as input the call type and the conversation, which are separated by a special token. Hence, we developed three distinct models based on the call type associated with the conversation during both training and testing. The discrepancy among these models is illustrated in Table 4.7.

	Train	Test
Pipeline$_C$ – oracle	∘	∘
Pipeline$_C$ – oracle/pred	∘	×
Pipeline$_C$	×	×

Table 4.7 – Difference between the different Pipeline$_C$ models. ∘: reference call type was taken. ×: predicted call type was taken .

— Pipeline$_C$ – oracle : This model takes the reference call type as input during the training and testing phase.

— Pipeline$_C$ – oracle/pred: We assume having the reference call type only for the train and the validation set. This means that the test set does not have any call type annotation. Therefore we used the classifier Conv-CT-classifier to predict the call type for each test set example.

— Pipeline$_C$: We assume no annotation exists for all the sets (none for training, validation, or testing). Then, we used the classifier Conv-CT-classifier to predict the call type for each dataset example.

System	R1	R2	RL	CT-Acc	NE-P	NE-R	NE-F1
Baseline	34.04	14.91	28.83	76.4	0.57	0.46	0.51
Multitask$_C$	33.59	14.23	28.42	76.4	0.54	0.19	0.28
Multitask$_E$	35.34	15.67	29.86	77.1	0.45	0.39	0.42
Pipeline$_C$ – oracle	34.65	15.10	29.33	73.6	0.47	0.45	0.46
Pipeline$_C$ – oracle/pred	34.46	14.93	29.09	73.6	0.47	0.45	0.46
Pipeline$_C$	34.09	14.79	28.99	76.4	0.49	0.47	0.48

Table 4.8 – Evaluation results of text summarization system. R1, R2, RL correspond to different ROUGE-Score. CT-Acc is the accuracy of call type prediction. NE-P, NE-R and NE-F1 is the entity precision, recall and F1 respectively.

We trained these different models as well as the baseline model defined in section 4.4.1. We computed the ROUGE score [Lin04] of each model to evaluate the ability of these models to generate n-grams close to the one in the reference summary and reported the results in Table 4.8. We also computed our proposed metric, **CT-Acc**, **NE-P**, **NE-R** and **NE-F1** to evaluate named entity and call type. Upon evaluation, the text summarization models ROUGE scores based on call types are comparable to the baseline. However, as demonstrated in [ZPR22], summaries generated from the DECODA dataset may contain unfaithful information such as hallucinations, misspellings, or omissions. Thus, we have chosen to conduct a manual evaluation of these summaries.

A comprehensive evaluation of the *informativeness* and *faithfulness* of summaries generated by baseline, Multitask$_C$, and Pipeline$_C$ has been done. We conducted a human evaluation of 30 randomly selected examples, considering a summary faithful if all its information could be traced back to the source document and informative if it covered the essential information. For this annotation, we considered the reference summary to be the most informative summary. Thus, to evaluate the informativeness, the annotator have compared the information of the generated summaries to the ones of the reference summaries. We ask one annotator to annotate each summary regarding these two criteria. The annotator followed these instructions:

- Read the source document and the different summaries
- Among the three systems summaries(baseline, Multitask$_C$, Pipeline$_C$), choose the one that is the more informative and the more faithful.
  - Write the name of the system that you rank as the best in terms of informativeness and faithfulness.
  - If you can not decide between the three, write *"undecided"*.
  - If the summaries are the same, write *same*.

The findings from the annotation process are displayed in Table 4.6. The table reveals that system Pipeline$_C$ was deemed faithful and informative 44% and 42% of the time, respectively. In contrast, system Multitask$_C$ performed the least effectively in both categories. It should be noted that these initial annotations require further

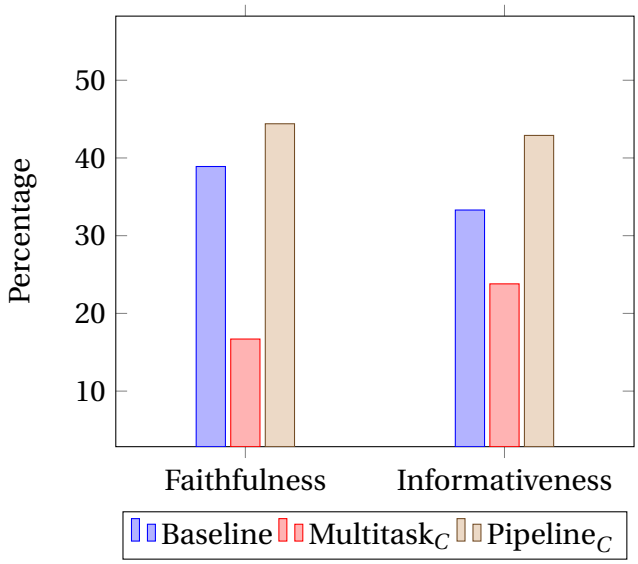


Figure 4.6 – Manual evaluation of the summaries generated by the baseline model, Multitask$_C$ and Pipeline$_C$. Percentage of times each system was ranked first in terms of faithfulness and informativeness.

confirmation from additional annotators to ensure the accuracy of the results, as inter-annotator agreement must be considered.

Summary selection using task information metric Our goal is to ensure that the predicted call types closely match the reference call types. In order to achieve this, we used a sampling method outlined in section 3.2.3.2 to generate a set of candidate summaries for each example. For each example, we generated an average of 40 summaries using different decoding strategies. As decoding strategies, we used Top-P, Top-K, Temperature, Beam and Greedy search. We set each parameter for the following values:

— Top-P: $P \in [0.70, 0.95[$ with the step of 0.05

— Top-K: $K \in [30, 100[$ with the step of 15

— Temperature: $T \in [0.7, 1]$ with the step of 0.1

— Beam decoding: We used a beam of size 6 and keep the 6-best.

All these values were taken empirically.

We selected the top two models from our previous human evaluation (baseline and Pipeline$_C$) to generate the summaries using sampling method. From these summaries, we chose the best one based on various criteria defined in section 4.4.2. The first criterion, called $\min D_{\mathrm{KL}}(G \parallel \mathrm{Conv})$, involves selecting the summary with the smallest KL divergence between the probability distribution of the prediction of call type from the conversation and the one from the generated summary. The second criterion, $\min \mathrm{NEHR} + D_{\mathrm{KL}}(G \parallel \mathrm{Conv})$, involves first selecting summaries with minimal NEHR, followed by choosing the one with the smallest KL divergence between the probability

distribution of the prediction of call type from the conversation and the one from the generated summary.

	R1	R2	RL	CT-Acc	NE-P	NE-R	NE-F1
Baseline model - BARThez fine-tuned to generate synopses							
BEAM	34.04	14.91	28.83	76.4	0.57	0.46	0.51
min $D_{\mathrm{KL}}(G \parallel \mathrm{Conv})$	33.99	14.23	28.13	**82.9**	0.54	0.43	0.48
min NEHR	34.12	14.13	27.79	73.6	0.58	**0.50**	**0.54**
min NEHR + $D_{\mathrm{KL}}(G \parallel Conv)$	33.70	14.57	28.04	82.1	**0.60**	0.45	0.51
Pipeline$_C$ - Baseline + call type as input, synopses as output							
BEAM	34.09	14.79	28.99	76.4	0.49	**0.47**	0.48
min $D_{\mathrm{KL}}(G \parallel \mathrm{Conv})$	33.85	14.60	28.87	**85.7**	0.53	0.42	0.47
min NEHR	34.18	14.69	28.52	75.7	0.55	**0.47**	**0.51**
min NEHR + $D_{\mathrm{KL}}(G \parallel Conv)$	33.32	14.48	28.58	84.3	**0.57**	0.45	0.50

Table 4.9 – R1,R2,RL: ROUGE; CT-Acc: Call type prediction from synopses (ref=oracle synopses); NE-R and NE-P: Entities recall and precision of the summaries generated compared to the reference

Table 4.9 shows the results obtained from applying various summary selection criteria. We used the ROUGE score along with the three semantic metrics to assess the quality of the generated summary and the semantic task information in the generated summaries.

We also reported the best BEAM hypothesis for each model in Table 4.9 to compare the different metrics. Table 4.9 shows comparable results regarding ROUGE scores for the different models. In contrast, we obtained a significant difference regarding the task information semantic scores. When we explicitly minimized the KL divergence between predicted call types from conversation and generated summary (min NEHR + $D_{\mathrm{KL}}(G \parallel Conv)$), we observed a marked increase in accuracy for model Pipeline$_C$. Combining the named entity hallucination risk (NEHR) minimization with the KL divergence on call types yielded a significant increase in entity precision (NE-P), accompanied by gains in call type accuracy (CT-Acc) and a slight decline in recall (NE-R).

4.4.5 Discussion

Our study delved into the effects of using task-semantic information in context of dialog summarization to have more faithful summary. We proposed two call-type based models and one entities-based model to enhance the semantic representation of generated summaries. While automatic evaluation metrics like ROUGE yielded comparable results to the Baseline, human evaluation revealed that Pipeline$_C$ surpassed the other systems (Baseline and Multitask$_C$) in terms of both fidelity and informativeness.

We found that using KL divergence between the generated summary and the source document as a summary selection criterion holds promise, as using the Syn-CT-classifier on the generated summary and compared to the reference call type led to an accuracy increase of approximately 10% (Call-type classification accuracy from the generated synopses - CT-Acc). However, certain call types were underrepresented in the dataset, indicating the necessity for further research to improve call type prediction and, correspondingly, the quality of the associated summary.

Combining KL divergence and NEHR (NEHR + $D_{\mathrm{KL}}(G \parallel Conv)$) as a selection criterion enhanced the entity precision of the model in comparison to the reference. Additional manual evaluation is needed to verify the efficacy of this combined criterion in relation to both fidelity and informativeness.

To sum up, we obtained promising results but we need to preform significance tests such as bootstrap or ttest to validate the different results as the ROUGE score are very close.

4.5 Data normalisation and data augmentation for spoken conversation summarization

4.5.1 Motivation

In section 4.3, the DECODA dataset was examined, revealing that only certain portions include annotated synopses. DECODA Part 1 summaries were composed in a non-literary manner, while DECODA Part 2 lacks annotated synopses. However, the dataset's quality and quantity are crucial for training a deep learning model that can generalize effectively. Furthermore, as noted in the section 4.2, both annotated summaries and conversation transcripts are required to train a conversation summary model using a pipeline approach. These situations are common in industrial settings, where conversations are frequently recorded but not transcribed or summarized. This is the motivation behind our study.

Our study aims to explore the impact of using automatic conversation transcription systems, as well as data normalization and augmentation techniques, on the faithfulness of generated summaries.

Based on our research, we have formulated three hypotheses. Firstly, we believe that *data normalization will lead to more faithful summaries that closely match the source document.* Secondly, we posit that *data augmentation will not decrease the quality of summaries generated by automatic systems.* Thirdly, we suggest that *using automatic transcription to train summarization models is a viable alternative to manual transcriptions, which are often challenging to obtain in industrial settings.*

Finally, we create a new DECODA dataset that we call *DECODA-summ-V2* with more annotated summary to perform dialog summarization.

4.5.2 Data normalisation and Data augmentation methods

We describe a realistic scenario where one has access to a corpus consisting solely of raw audio conversations, without any annotations or transcriptions, but with partial annotation of the summaries. The summaries consist in bullet point summaries that are not written in a literary style. The second realistic scenario is having access to only the raw audio without any annotations or transcriptions, except for a small subset with target summaries written in literary style. We show how generic tools for speech transcription and automatic summarization powered by LLM can be leveraged for data normalisation (in the case there is a bullet point summary) and data augmentation (in the case there is a small subset with a target summary).

4.5.2.1 Automatic transcription for text summarization

As manual transcriptions are difficult to acquire without annotation, the use of ASR systems is a good alternative. Thus, we used WhisperX [Bai+23], a state-of-the-art model for ASR based on Whisper [Rad+22]. Whisper is an ASR system trained on a large and diverse dataset, enabling transcription in multiple languages and translation into English. It uses a simple end-to-end approach, implemented as an encoder-decoder Transformer to facilitate smooth and easy transcription. The decoder generates highly accurate text captions using specialized tokens that enable the model to perform complex tasks such as language identification, timestamps, and speech-to-English translation. WhisperX adds three components to Whisper. It uses Voice Activity Detection (VAD) to segment input audio, divides it into chunks with minimally active speech regions, and implements forced alignment for precise timestamps.

We decided using WhisperX as it is a state-of-the-art approach for ASR and it does not need to be refined on our data. Thus, it can be used in practical context where it is a challenge to have access to manual transcription. WhisperX has 5 different models according to size. Table 4.10 gives an example of manual transcription vs automatic transcription with the biggest WhisperX model of DECODA dataset.

For our experiment, we used all the model sizes of WhisperX in order to evaluate the impact of ASR on dialog summarization.

4.5.2.2 Normalize dialog summarization dataset

Within section 4.3, we provided an overview of the DECODA dataset and the contrast between its first and third parts. Specifically, the third part offers more elaborate, literary-style summaries, while the first part serves as a summary with less detail presented as bullet points. This is a common challenge as literary-style annotations with a lot of detail are difficult to achieve.

To address this issue of data homogeneity, we decided to standardize the first part of the DECODA dataset to match the literary style of the third part. Our approach involved utilizing LLM, which are pre-trained on vast corpora of text data and can perform a wide range of NLP tasks, including text generation, translation, summarization, and question answering, among others. Over recent years, a variety of LLM have

MANUEL TRANSCRIPTION: *autre :* NNAAMMEE va vous répondre *conseiller :* NNAAMMEE bonjour *appelant :* bonjour monsieur *appelant :* monsieur j' ai un renseignement à vous demander *conseiller :* oui *appelant :* j' ai grand garçon handicapé mental qui a une un coupon Améthyste *conseiller :* hm *appelant :* bon évidemment ça ne dure pas un an il se démagnétise et avant j' allais euh trente rue Championnet *conseiller :* oui *appelant :* le faire changer *conseiller :* hm *appelant :* mais leur numéro de téléphone a changé auriez vous l' amabilité de lu de me donner le numéro le nouveau numéro *conseiller :* alors VGC à Championnet c' est le zéro un cinquante-huit *appelant :* oui *conseiller :* soixante-dix-sept trois fois *appelant :* soixante-dix-sept trois fois *conseiller :* hm *appelant :* très bien je vous en remercie *appelant :* monsieur *conseiller :* bonne *conseiller :* journée madame au revoir *appelant :* au revoir
AUTOMATIC TRANSCRIPTION: *[SPEAKER_00]:* On va vous répondre. *[SPEAKER_00]:* Bonjour. *[SPEAKER_01]:* Bonjour monsieur. *[SPEAKER_01]:* Monsieur, j'ai un renseignement à vous demander. *[SPEAKER_01]:* J'ai un grand garçon handicapé mental qui a un coupon améthyste. *[SPEAKER_01]:* Bon, évidemment, ça ne dure pas un an. *[SPEAKER_01]:* Il se démagnétise. *[SPEAKER_01]:* Et avant, j'allais 30 russes championner, le faire changer. *[SPEAKER_01]:* Mais leur numéro de téléphone a changé. *[SPEAKER_01]:* Auriez-vous l'amabilité de me donner le nouveau numéro ? *[SPEAKER_00]:* Alors, VGC à Championnet, c'est le 0 en 58. *[SPEAKER_00]:* Oui. *[SPEAKER_00]:* 77, 3 fois. *[SPEAKER_01]:* 77, 3 fois. *[SPEAKER_01]:* Très bien. *[SPEAKER_01]:* Je vous en remercie, Monsieur. *[SPEAKER_00]:* Bonne journée, Madame. *[SPEAKER_00]:* Au revoir. *[SPEAKER_01]:* Au revoir.

Table 4.10 – Manual transcription vs Automatic transcription with Whisper large (Example 20091112_RATP_SCD_0152 of DECODA).

emerged, such as GPT [Bro+20; Ope+24], Llama [Tou+23b; Tou+23a], Mistral [Jia+23], and several others.

Thanks to their architecture and training, LLM can be adapted to a wide range of natural language processing tasks. They are incredibly versatile and capable of performing various tasks without requiring extensive retraining.

For our cases, we made the decision to utilize the state-of-the-art language model, ChatGPT-3.5, to normalize the first part of DECODA. This decision was based on the model's remarkable ability to generate coherent and fluent text in French, while also being able to perform various tasks without the need for extensive finetuning. The methodology involves:

1. Rewriting bullet point summaries in a literary style with an instruction-based LLM in a few-shot learning mode. Feeding the system a few examples of target summaries, and prompting it to generate summaries that resemble these targets.
2. Fine-tuning a text summarization model on the dataset including the rewritten summaries and the human ones.

This methodology can be adapted to any situation with such kind of dataset.

For normalization, we tasked LLM with rephrasing the current summary via the given prompt:

> *Réécris de manière plus précise et en français correct le texte suivant qui décrit un dialogue entre #APPELANT, usager des transports parisien, et #CONSEILLER de la RATP :*

The one-shot example used is as follows:

> *Entrée : "état du trafic bus 47 car 1 bus sur 4 constaté, en effet grève dépôt de Vitry, contact régulateur, pas mal de bus mais le prochain dans ce sens est loin, conseil de prendre le métro. Locuteur typiquement parisien",*
> *Sortie : "Un appelant souhaite connaître l'état du trafic du bus 47, car il constate qu'il n'y a qu'un seul bus sur quatre prévu. La conseillère lui confirme qu'il y a effectivement une grève au dépôt de Vitry. Après avoir contacté le régulateur elle lui conseille de prendre le métro car le prochain bus annoncé est lointain.*

Table 4.11 give an example of normalize summary using ChatGPT-3.5.

Here we consider having a bullet point summary. What if we do not have annotation for a part of how dataset ?

DIALOG: *autre :* NNAAMMEE va vous répondre *conseiller :* NNAAMMEE bonjour *appelant :* bonjour monsieur *appelant :* monsieur j' ai un renseignement à vous demander *conseiller :* oui *appelant :* j' ai grand garçon handicapé mental qui a une un coupon Améthyste *conseiller :* hm *appelant :* bon évidemment ça ne dure pas un an il se démagnétise et avant j' allais euh trente rue Championnet *conseiller :* oui *appelant :* le faire changer *conseiller :* hm *appelant :* mais leur numéro de téléphone a changé auriez vous l' amabilité de lu de me donner le numéro le nouveau numéro *conseiller :* alors VGC à Championnet c' est le zéro un cinquante-huit *appelant :* oui *conseiller :* soixante-dix-sept trois fois *appelant :* soixante-dix-sept trois fois *conseiller :* hm *appelant :* très bien je vous en remercie *appelant :* monsieur *conseiller :* bonne *conseiller :* journée madame au revoir *appelant :* au revoir
SUMMARY: numéro VGC pour faire changer carte Améthyste démagnétisée
NORMALISATION: L'appelante demande le numéro VGC pour faire remplacer une carte Améthyste qui a perdu son champ magnétique.
AUGMENTATION: Une personne appelle pour demander le nouveau numéro pour changer un coupon Améthyste pour son fils handicapé mental. Le conseiller fournit le nouveau numéro de contact pour la demande.

Table 4.11 – Data augmentation and data normalisation example (Example 20091112_-RATP_SCD_0152 of DECODA).

4.5.2.3 Create summary using automatic system

In case we have no summaries annotation for our data (as for DECDOA part 2), we can rely also on LLM models to perform data augmentation to train the dialog summarization model. The methodology involves:

1. Use the manual transcription if you have access or applying a generic speech transcription system across the entire conversation corpus to produce textual transcriptions.
2. Processing these transcriptions with an instruction-based LLM in a few-shot learning mode, feeding the system a few examples of target summaries, and prompting it to generate summaries that resemble these targets.
3. Fine-tuning a summary generation system on the dataset including both the automatically generated summaries and the target summaries.

To facilitate data augmentation, we employed one-shot learning to assist the model in its generation. Specifically, we provided an example of a human summary from Part 3 and prompt the model to generate a summary for various input transcriptions based on the prompt :

> *Résume en français correct de manière concise, sans donner beaucoup de détails, le texte suivant qui représente un dialogue entre #APPELANT qui utilise les transports parisien et #CONSEILLER de la RATP. Chaque tour de parole commence par le nom de la personne qui parle entre crochets, puis la transcription du tour. Chaque tour se termine par le symbole <END>. Voici la transcription à résumer :*

The one-shot example used is as follows:

> QUESTION : "*[null] va vous répondre <END> [conseiller] bonjour <END> [appelant] oui bonjour madame <END> [appelant] madame je voudrais savoir est-ce que demain matin il est prévu des perturbations sur la ligne B euh du RER en direction euh de Roissy - Charles-de-Gaulle <END> [conseiller] alors les perturbations non sur le RER B c'est que le soir hein au niveau de l'aéroport mais pas dans la journée <END> [appelant] ah ah il y a des demain soir il y a des euh perturbations prévues ? <END> [conseiller] euh oui parce que vous avez des travaux sur le tronçon <END> [conseiller] SNCF donc <END> [appelant] ah bon sans ça <END> [appelant] ça marche normalement ? <END> [conseiller] donc si vous voulez c'est pour ça que ça mais si vous prenez vous vous partez à quel moment ? <END> [conseiller] le matin ? <END> [appelant] oh <END> [appelant] très tôt <END> [conseiller] alors dans ce cas-là non <END> [appelant] euh ve vers 6 heures du <END> [appelant] matin <END> [conseiller] non non il y a pas de problème <END> [appelant] il y a pas de problème <END> [conseiller] non non non non <END> [appelant] c'est sur le E uniquement qu'il y a des perturbations ? <END> [conseiller] alors sur le E oui vous avez d oui oui tout à fait où là c'est euh c'est aussi en travaux <END> [conseiller] il y a beaucoup de <END> [appelant] <END> [conseiller] travaux actuellement oui <END> [appelant] bon ! autrement dit euh p demain oh euh pas de problème sur le B <END> [conseiller] hm non <END> [appelant] merci beaucoup <END>*

[conseiller] mais je vous en <END> [conseiller] prie au revoir monsieur <END> [appelant] au revoir madame <END>.",
ANSWER : "*Un appelant souhaite savoir si la Ligne B du RER sera perturbée le lendemain, en raison des travaux sur le trajet. La conseillère indique que des perturbations sont prévues le lendemain, mais uniquement dans la soirée.*"

Table 4.11 gives an example of summary generated using ChatGPT-3.5

Creating a dedicated summary system is more practical and cost-effective than using a generic LLM. It provides better control over output and allows for more sophisticated strategies, like combining automatic speech understanding and summarization, as discussed in the following section.

4.5.3 Experiments and results

The purpose behind this study is to conduct a thorough analysis of the potential impact of utilizing ASR systems, such as WhisperX and ChatGPT-3.5, to obtain annotated data on the overall quality of the summary generated from systems that have been trained on these data. This study aims to delve deeper into the effectiveness of these systems and their ability to provide annotated data that can be used to enhance the train a dialog summarization system.

4.5.3.1 Dataset

We used the DECODA corpus [Bec+12], which is detailed in section 4.3 and is composed of three parts: Part 1, containing bullet-point summaries, Part 2, with no synopses, and Part 3, featuring detailed synopses written in a literary manner. In section 4.5.2, we discuss our decision to keep only a small number of human synopses used for system training. The remaining synopses were either normalized (for Part 1 synopses) or augmented (for Parts 1 and 2 synopses). Part 3 consisted of 500 dialogs, with 200 used for model evaluation (test set), 200 for system fine-tuning (train-H for human annotated data), and 100 for validation and parameter adjustment. In addition to the gold summaries, we created three corpora:

— Train-B (for *bullet*): we took the bullet point synopses of part 1 to form a corpus;

— Train-N (for *normalization*): We used prompt-based LLM from OpenAI (ChatGPT-3.5) to make existing synopses of part 1 DECODA more literary

— Train-A (for *augmented*): We used prompt-based LLM from OpenAI (ChatGPT-3.5) on the transcripts of parts 1 and 2 DECODA corpus.

Table 4.12 provides statistics for the training and test sets, revealing the longer length of synopses in the test set compared to those in the Train-H+N and Train-H+A sets. This difference occurs because the augmented data does not consistently match the style of human summaries in DECODA due to its 1-shot approach. Train-H+B synopses are shorter than Train-H as Part 1 synopses are synthetic.

Statistics	Train-H	Train-H+B	Train-H+N	Train-H+A	Test
# dialogs	200	697	697	1390	200
# avg dialog size	545.0	459.3	459.3	470.3	495.6
# avg synopsis size	55.3	29.4	42.1	47.9	52.7

Table 4.12 – Decoda data distribution in the train (human, human + bullet point, human + normalisation, human + augmentation) and the test set

4.5.3.2 Models parameters

We used BARThez [KTV21] as in section 4.4.4.2. For consistency with previous experiments, we set the maximum sizes of conservation and synopses at 1024 and 128, respectively. Our models were trained on Train-H+B, Train-H+N, and Train-H+A datasets for 15 epochs using AdamW optimizer with a learning rate of 5×10^{-5}. We also trained a model on Train-H for 10 epochs. Only models that minimized loss on the validation set were saved. To accommodate the lack of speaker identification in automatic transcription, we removed the speaker IDs and preserved the raw text with speech turns separated by line breaks.

4.5.3.3 Impact of data normalisation and augmentation

To evaluate the impact of using data normalization or data augmentation for dialog summarization, we trained different models on the various training data sets presented in Section 4.5.3.1. We then generated summaries from our test set to compare model performance. The ROUGE and BERTScore for each model were computed and reported in Table 4.13. Additionally, we recorded the results obtained when applying the model trained for text summarization on Orange-sum to DECODA data without prior fine-tuning in the same table. To complete our analysis, we integrated the ROUGE and BERTScore performance of ChatGPT3.5 on the test data into Table 4.13.

System	Train data (size)	R1	R2	RL	BS	CT-Acc	NE-P	NE-R	NE-F1
Barthez	OrangeSum (0)	22.77	5.87	14.36	7.82	0.61	0.49	0.33	0.39
Barthez	train-H (200)	33.22	13.22	23.59	33.12	0.77	0.49	0.49	0.49
Barthez	train-H+B (697)	38.40	15.48	27.64	37.87	0.75	0.51	0.49	0.49
Barthez	train-H+N (697)	39.07	15.79	27.65	37.50	0.76	0.52	0.50	0.51
Barthez	train-H+A (1390)	40.40	16.96	29.11	38.90	0.77	0.54	0.48	0.51
chatGPT	one-shot (1)	40.59	15.95	28.93	37.30	0.80	0.64	0.42	0.51

Table 4.13 – RougeL and BERTScore on the test partition on the manual transcription of DECODA with Barthez fined tuned on the different datasets (Orange-Sum [KTV21], train-H, train-H+B, train-H+N, train-H+A). Performance of chatGPT3.5 is also presented. (size) refers to the number of DECODA example each system seen during training.

The model trained on Orange-Sum and evaluated on our test set created from DECODA produces unsatisfactory results. However, its performance is normal, as it was specifically trained for text summarization and not for conversational tasks. The structural and contextual differences between conversation and text make it challenging for the model to perform better without prior training.

The divergent outcomes of training the model on Train-H and Orange-Sum datasets provide crucial insights into the complexities of text and dialogue summarization. This comparison illuminates the limitations of text summarization systems when applied to dialogue summarization tasks without prior fine-tuned.

Incorporating part 1 into the train-H training data yields an improved ROUGE score and BERTScore, even though the summaries in part 1 do not match the style of those in part 3, which were used as test data.

The results of data normalization show that the generated summaries are comparable to the non-literary written summaries in Part 1. This is because our test data originates from the literary summaries (Part 3), and the model when trained with some examples from Part 3, can produce summaries in the style of Part 3. While normalizing the summaries of Part 1 may not make a significant difference, adding them as training data brings a considerable enhancement, as evidenced by a 4-point increase in ROUGE-L score. Incorporating summaries generated by LLMs from Part 1 and 2 in Part 3 (train-H+A) data yields better results than training with only train-H dataset data using the Barthez model, with an almost 2-point increase in ROUGE score. One-shot learning with ChatGPT3.5 produces results similar to those of data augmentation in terms of R1. But gives high semantic score as the CT-Acc and the en-P is higher.

4.5.3.4 Impact of WER on summarization performance

We evaluated different sizes of WhisperX and computed its Word Error Rate (WER) score based on manual transcriptions. The WER score is computed as follow:

$$WER = \frac{S + D + I}{N} \tag{4.6}$$

Where:

S = Number of substitutions
D = Number of deletions
I = Number of insertions
N = Number of words in the reference transcription

We used Barthez [KTV21] to conduct a dialog summarization task, using transcriptions from different WhisperX sizes to analyze the effect of the WER score on the summarization score such as ROUGE [Lin04] and BERTScore [Zha+20c]. The results are consigned in Table 4.14.

Regardless of the system ASR applied, the performance of systems trained on its transcriptions falls short of that of the summarization system trained on manual

Transcript	WER	R1	R2	RL	BS	CT-Acc	NE-P	NE-R	NE-F1
Manual	0.00	40.40	**16.96**	**29.11**	**38.90**	0.77	**0.54**	**0.48**	**0.51**
Whisper (T)	76.37	35.01	11.52	24.53	34.38	0.75	0.35	0.28	0.31
Whisper (B)	58.63	37.43	13.21	26.22	35.88	0.75	0.39	0.30	0.34
Whisper (S)	46.37	38.40	14.45	26.99	36.95	0.75	0.49	0.35	0.41
Whisper (M)	40.87	38.75	14.52	27.60	36.82	**0.79**	0.51	0.35	0.42
Whisper (L)	40.02	39.26	14.92	27.67	37.14	0.76	0.53	0.36	0.43

Table 4.14 – WER score on automatic transcription from different size of WhisperX: T-Tiny, B-base, S-small, M-medium, L-large. Rouge-L and BERT-scores on summaries generated by Barthez finetuned and evaluated on automatic transcriptions. CT-Acc, NE-P, NE-R and NE-F1 our semantic measure based on call type and named entities.

transcriptions. We observed that as the WER score decreases, both the ROUGE and BERT scores improve. Our findings indicate that the WER score, i.e. the quality of the transcription, plays a role in dialog summarization.

4.5.4 Discussion

We conducted a study to assess the impact of data augmentation and data normalization on summary generation. Our approach involved training a model on various data configurations and testing it on a test set that exclusively contained DECODA part 3 that is more detailed and written in a literary style.

Our findings revealed that data normalization did not enhance the values of our evaluation metrics. However, we obtained slightly higher scores with data augmentation, although we still need to conduct significance tests to validate our results. Nevertheless, the increase in data homogeneity with the same literary style could prove beneficial in many cases. Indeed, this new dataset will enable more in-depth studies of task-oriented dialog summarization by the community.

For instance, Table 4.11 illustrates how the basic summary fails to convey how the task was resolved. On the other hand, the summary generated by ChatGPT-3.5 provides this crucial information. This begs the question: why not use ChatGPT-3.5 as a conversation summarization system?

The primary advantage of a streamlined system is twofold: it can improve results for a specific task and is less resource-intensive. While ChatGPT-3.5 may outperform BARThez on some semantic scores, such as CT-Acc and NE-P, it is as cost-effective. BARThez has 216 million parameters, while ChatGPT-3.5 has 175 billion. Even the smallest LLM has 7 billion parameters, making BARThez a more practical choice.

Regarding automatic transcription, it is worth noting that WhisperX presents a viable alternative to the shortage of manually annotated transcriptions and the associated costs. The outcomes of the models suggest that this approach could be a promising substitute.

In the following section, we put ourselves in the position of not having access to manual transcription. Experiments are made on the automatic transcription generated by WhisperX large as the ROUGE and BERTScore are the closest to the results obtained for manual transcription. We went to study the impact of integrating task-semantic information to text summarization when we increase the dataset and used ASR system on the raw audio.

4.6 SLU and data augmentation for dialog summarization

We assume that integrated task-semantic information in the text summarization process may help fill the gap between system trained on manual transcription and the one using automatic transcription. Thus, we decided to apply our previous experiments in section 4.4 on our new dataset *DECODA-summ-V2*.

4.6.1 Integrate call type and named entities

With *DECODA-summ-V2* dataset, we delved into the efficacy of integrating semantic task information in model training. One new multitask framework based on NEs was introduced in addition to those presented in section 4.4.1. Let D be the input dialog, S the generated summary, C the call type, and E the set of entities appearing in the summary. We modified the **Multitask**$_C$ framework to integrate named entities. Thus, we used the following:

1. **Baseline**: Remains unchanged
2. **Pipeline**$_C$: Remains unchanged
3. **Multitask**$_{\{C,E,CE\}}$: Generate both semantic labels such as call type or entities, and summary directly from the conversation transcript, resulting in three systems generating the call type, the entities or both prior to generating the summary.

$$\{C, E, CE\}, S = \text{summary} \circ \text{semantics}(D)$$

Figure 4.7 gives more inside into the integration of semantic information about the task like named entities and call types.

We also reproduced the experiment made using NEHR and as a summary in *DECODA-summ-V2* dataset (section 4.4.2) in order to see the contribution of the proposed methodology.

4.6.2 Experiments and results

4.6.2.1 Experimental setup

We trained two classifiers on *DECODA-summ-V2* dataset as in section 4.4.4.2 based on CamemBERT [Mar+20]: **Conv-CT-classifier** that take the conversation and predict

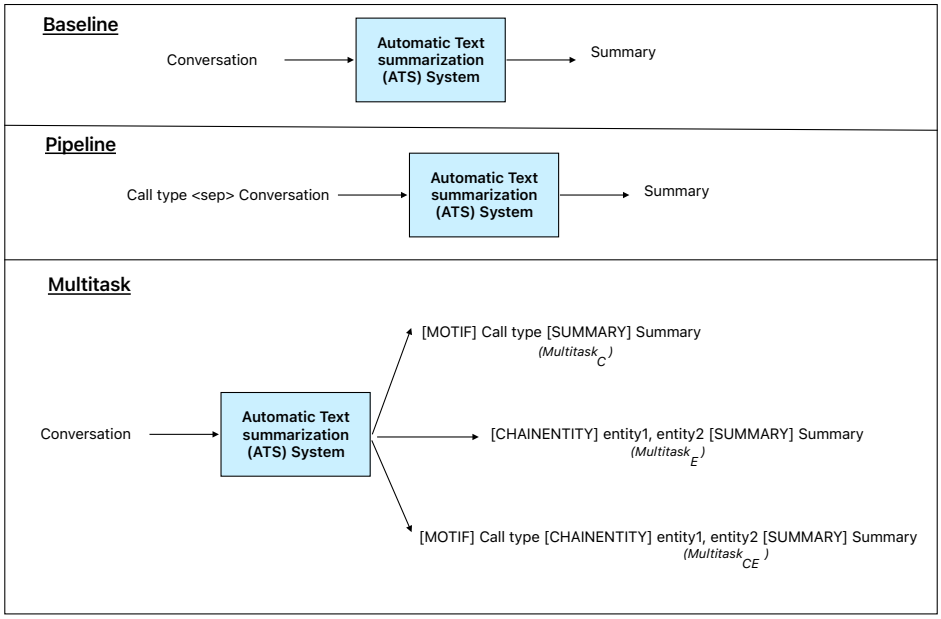


Figure 4.7 – Integration of semantic information about the task during summary generation process

the call type, and **Syn-CT-classifier** that identifies call type from synopsis. For summarization system, we used the same pre-trained model BARThez [KTV21] and training parameters than the one used in previous section.

4.6.2.2 call type classification results

We trained the two call type classifiers Conv-CT-classifier and Syn-CT-classifier. Conv-CT-classifier takes as input the automatic transcription from WhisperX large to predict call type, Syn-CT-classifier takes synopses as input to predict call type. Both models were trained on the Train-H+A dataset which is the training dataset with augmented synopses (Part 1 and Part 2 synopses were generated with ChatGPT3.5), and evaluation were done on the test dataset which contains only DECODA Part 3 examples.

We evaluate the classifiers in terms of accuracy and the weighted f1 score by comparing the predicted call type to the reference one. The results are recorded in Table 4.15.

The results from both classifiers are inconsistent with the previous section's findings (4.4.4.3), where the conversation classifier outperformed the synopsis classifier. Indeed, here Conv-CT-classifier classifier gives the same result as Syn-CT-classifier. This similarity could be due to the use of data augmentation during model training. In the previous experiment (4.4.4.1), we used gold synopses from part 1 which are synthetic, combined with some from part 3, for model training. Unfortunately, this resulted in

System	Acc.	W-F1
Conv-CT-classifier	81	80
Syn-CT-classifier	80	80
Multitask$_C$	77	75
Multitask$_{CE}$	73	71

Table 4.15 – Accuracy and Weighted F1 Score of call types classifiers and generation systems

synopses that were inadequate in providing sufficient information for the classifier to make precise decisions. However, by incorporating LLM-generated synopses for parts 1 and 2, we obtained synopses with a style closely resembling those of part 3. As a result, the classifier performed with the same degree of accuracy for conversations as well as summaries.

We also recorded in table 4.15 the performance of the multitasks Multitask$_{C,CE}$ which consist in generating call type before generated synopses for Multitask$_C$ and call type followed by NEs and synopses for Multitask$_{CE}$. Multitask$_C$ performed better than Multitask$_{CE}$. This can be due to the various information Multitask$_{CE}$ have to generate. Our hypothesis is that the model's multitasking ability may decrease its performance on specific tasks, such as generating call types, named entities, and summaries.

4.6.2.3 Impact of task semantic information during training

We trained pipeline and multitask models (detailed in section 4.6.1) that incorporate semantic conversation information on the train-H+A dataset (an augmented training dataset) with transcription from WhisperX large and evaluate each model on the testing dataset. The results, presented in table 4.16, include automatic summarization scores like BERTScore [Zha+20c] and ROUGE score [Lin04]. Additionally, we calculated the CT-Acc score, which measures the accuracy between reference call types and those predicted by Syn-CT-classifier for each generated summary. We also provided entity precision comparisons between the reference and generated summaries.

System	R1	R2	RL	BS	CT-Acc	NE-P	NE-R	NE-F1
Baseline	39.26	14.92	27.67	37.14	0.76	0.53	0.36	0.43
Pipeline$_C$	39.34	14.82	27.61	36.97	**0.79**	**0.54**	0.36	0.43
Multitask$_C$	38.98	14.91	27.41	37.16	0.76	0.51	0.35	0.42
Multitask$_E$	39.09	14.94	28.03	37.21	0.75	0.46	0.35	0.40
Multitask$_{CE}$	39.24	14.76	27.51	37.02	0.74	0.51	0.38	0.44

Table 4.16 – Summaries evaluation. R1, R2, R3 are ROUGE Score, BS is BERT-scores. CT-Acc, NE-P, NE-R, NE-F1 are our proposed semantic information.

Our Pipeline$_C$ model improved CT-Acc on call type while maintaining similar results in terms of ROUGE-L and EN-prec. Surprisingly, our Multitask$_{CE}$ model did not improve NE-Prec, which may be attributed to the complexity of the task, as it had low accuracy scores in generated call types. However, our Multitask$_C$ and Multitask$_E$ models yielded comparable results to the baseline. This suggests that integrating semantic information such as call types and named entities for multitask generation may not significantly enhance the semantic quality of the generated summary.

What about using the task semantic information as a summary selection criterion for augmented data?

4.6.2.4 Impact of task semantic information as a summary selection

In section 4.4.4.3, using the $D_{\text{KL}}(G \parallel \text{Conv})$ or NEHR for selecting a summary resulted in improved scores for the dialogue's semantics, particularly the CT-Acc and NE-Prec metrics. These metrics measure the correctness and precision of the generated summary's call type and named entities, respectively, compared to the reference summary. As a result, we opted to conduct a similar experiment to determine if selection can enhance summary semantics in relation to the task, using augmented data.

As Pipeline$_C$ increased the CT-Acc, we used the model trained to generate various summaries using sampling decoding strategies and select the one according to the selection metrics presented in section 4.4.2: min $D_{\text{KL}}(G \parallel \text{Conv})$ and min NEHR.

System	R1	R2	RL	BS	CT-Acc	NE-P	NE-R	NE-F1
Pipeline$_C$								
BEAM	39.34	14.82	27.61	36.97	0.79	0.54	0.36	0.43
min $D_{\text{KL}}(G \parallel \text{Conv})$	38.91	14.58	27.77	36.84	**0.82**	0.52	0.38	0.44
min NEHR	40.51	15.12	27.65	36.36	0.81	**0.58**	0.38	**0.46**
min NEHR + $D_{\text{KL}}(G \parallel Conv)$	39.18	14.86	27.89	37.01	**0.82**	0.51	0.39	0.44
ChatGPT3.5								
Temp=1	38.72	13.87	26.62	35.41	0.79	0.63	0.36	0.46
min $D_{\text{KL}}(G \parallel \text{Conv})$	38.35	13.26	26.11	34.35	**0.83**	**0.65**	0.37	0.47
min NEHR	38.34	13.44	26.37	34.80	0.80	0.60	0.40	**0.48**
min NEHR + $D_{\text{KL}}(G \parallel Conv)$	39.66	15.94	28.19	37.24	**0.83**	0.47	0.44	0.46

Table 4.17 – Summaries evaluation of summary selection method. ChatGPT3.5 results are included

We report the results in Table 4.17 and denote as **Pipeline**$_C$ - min $D_{\textbf{KL}}(G \parallel \textbf{Conv})$ for summaries selected using the D_{KL} criterion and **Pipeline**$_C$ - min **NEHR** for summaries selected using NEHR criterion. According to the findings, we observe a 3-point increase in the CT-Acc for Pipeline$_C$ when using D_{KL} as the selection criterion (Pipeline$_C$ - min D_{KL}), and a 4-point increase in the NE-Prec when using NEHR as the criterion (Pipeline$_C$ - min NEHR). However, these enhancements are relatively modest in comparison to the outcomes achieved in the prior experiment (refer to section 4.4.4.3).

We used ChatGPT3.5 to generate several summaries and assess the performance of LLM. ChatGPT3.5, D_{KL}-based selection yields the best results in terms of CT-Acc and entity precision. These results suggest that the choice of selection criteria could have different semantic impacts on small models versus LLM, and additional experiments are needed to validate this hypothesis.

We also decided to evaluate the contribution of summary selection for a model trained on automatic transcriptions with a high WER score. Thus, we used the transcription of WhisperX base model to train the summarization model and apply the methodology for summary selection. Results are consigned in Table 4.18.

The ROUGE scores are similar with the exception of the $\min D_{\mathrm{KL}}(G \parallel \mathrm{Conv})$ criterion selection which has a low score. This is because the selection criterion involves matching call type of conversation to the one of generated summaries. Given that the conversation includes numerous errors due to low WER score, selecting a summary based on this criterion can result in summaries with lower ROUGE scores when compared to the reference. By using NEHR-based selection, the semantic score on named entities is enhanced without compromising the ROUGE score. The use of these two selection metrics yields a noteworthy improvement in scores pertaining to named entities and a modest improvement in those related to the call type.

System	R1	R2	RL	BS	CT-Acc	NE-P	NE-R	NE-F1
BEAM	37.43	13.21	26.22	35.88	0.75	0.39	0.30	0.34
min NEHR	37.11	12.99	25.67	35.52	0.75	0.45	0.40	0.42
$\min D_{\mathrm{KL}}(G \parallel \mathrm{Conv})$	34.98	12.31	24.77	32.96	0.79	0.40	0.31	0.35
$\min \mathrm{NEHR} + D_{\mathrm{KL}}(G \parallel Conv)$	36.58	12.93	25.53	35.48	0.77	**0.46**	**0.43**	**0.44**

Table 4.18 – Summaries evaluation of summary selection method using automatic transcription provided by WhisperX base.

4.6.3 Discussion

In this section, we conducted an analysis of the effectiveness of integrating task-specific information into our proposed dataset's summarization system. Our findings indicate that including semantic information during training did not lead to noticeable improvements. However, we did observe a slight enhancement in summary selection when evaluating task-related semantic measures with KL divergence. This could be due to the already effective capturing of task-related semantic information by automatic transcripts and LLM-generated annotated summaries. Furthermore, we tested this approach on a dataset that uses the WhisperX base model as a transcription model and found that minimizing the NEHR resulted in improved named entities generation.

In a summary generated through automated transcription with a high WER score, selecting summaries based on semantic criteria such as NEHR and $\min D_{\mathrm{KL}}(G \parallel \mathrm{Conv})$ results in a superior summary with regards to named entities. However, the validity of this hypothesis requires human evaluation of the selected summaries.

4.7 Conclusion

In this section, we have explored several hypotheses related to task-specific semantic information and its impact on the quality of text summarization models. This was evaluate using traditional metric for text summarization to which we have added semantic measures. Indeed, we proposed measure based on task semantic information such as call type and named entities. Our findings suggest that incorporating semantic information into the system's input or output or during the summary selection process can lead to more accurate and faithful summaries from a task-specific perspective. Additionally, the selection of summaries according to these semantic criteria results in better summaries from the point of view of task semantics.

Furthermore, we proposed a new dataset, *DECODA-summ-V2* for task-oriented dialog summarization that we designed using LLM model which we make available to the community for advanced research. The dataset include also automatic transcription from the state-of-the-art model WhisperX for further investigation. By analyzing this new dataset, we found that increasing the amount of data and reducing the WER score of the automatic transcription systems can improve the quality of summaries.

Our contribution can be summarized as :

— Proposal of four news criteria to evaluate the semantic of the generated summaries.

— Proposal of a new version of the DECODA dataset (*DECODA-summ-V2*) with larger and homogeneous data.

— Proposal of two selection criteria based on task-semantic information.

— Proposal of multitask and pipeline method based on task-semantic information.

Overall, our results provide valuable insights into the importance of incorporating task-specific semantic information and increasing the amount of data when training text summarization models. These findings can help researchers and practitioners develop more effective and efficient summarization systems in the future.

Several studies are still needed to validate our findings, including assessments by humans of the generated summaries and conducting significance tests. Additionally, this section solely concentrated on obtaining semantic information such as named entities or call types, which were easily accessible in the corpus used. However, we aim to investigate other semantic information, such as call flow annotations or semantic frames, to ensure the accuracy of the generated summaries. We are confident that exploring diverse semantic perspectives will produce summaries that are faithful to the original document.

As we focus in this section only on pipeline method, exploring the end-to-end method is one of our future concerns. Indeed, this will avoid the need to switch to automatic transcription systems for the summarization task.

▷ [Aka+24a] Approche multitâche pour l'amélioration de la fiabilité des systèmes de résumé automatique de conversation - **Eunice Akani**, Benoit Favre, Frederic Bechet, and Romain Gemignani. 2024. JEP-TALN-RECITAL
https://aclanthology.org/2024.jeptalnrecital-taln.23/
▷ [Aka+24b] Unified Framework for Spoken Language Understanding and Summarization in Task-Based Human Dialog processing - **Eunice Akani**, Frederic Bechet, Benoit Favre, and Romain Gemignani. 2024. Interspeech (accepted)

Table 4.19 – List of papers based on this chapter

Conclusion

Our thesis embarked on a novel exploration of the faithfulness of summaries generated to the source documents. Indeed, The faithfulness of the generated summaries regarding the source document is crucial. A summary that strays from the original content is not valuable and fails to serve its purpose effectively, particularly in today's era when we are inundated with copious amounts of information to digest. Actually, we need a consistent summary of all this information.
We delved into two distinct sources of data: written text and transcripts of spoken conversations. Our central research question revolved around ensuring the faithfulness of generated summaries to their source. To tackle this, we undertook a comprehensive investigation, dividing our study into two distinct parts. The first part focus on text summarization while the second one on task-oriented dialog summarization.

We began by studying in Chapter 2, the errors found in generated summaries and analyzed the abstractive nature of the reference summaries. Thus, we trained four different models to generate summaries and analyze the different outputs regarding the error typology. Our analysis on Orange-Sum corpus reveals that systems based on pre-trained language models generate information that is not supported by the source document. Regarding the reference summaries, we found that most abstractions are not inferable using the document or the annotator's knowledge. This result can be explained by the nature of the reference summaries which contain information that is not present in the source documents (as news headers typically include pieces of information not discussed in the text body). With that kind of training data, models are trained to generate out-of-document information. Then, we focus on named entities which we hypothesize to be a good representation of a text and crucial information for fully understanding. Based on our findings, we hypothesized that having entities outside of the source document led to hallucinations. Our study confirmed this hypothesis, as we discovered that text summarization systems, especially systems based on pre-trained language models, mostly generated hallucinated entities when they were outside the source document. We also discussed the relevance of the news corpora for the text summarization task as their summaries contain out-of-document information.

Based on our analysis in Chapter 2, we proposed in Chapter 3 to reduce the number of hallucinations on named entities. Our study was based on the hypothesis that having fewer named entities outside the source document reduces hallucination. Thus, we introduced NEHR, a new criterion based on named entities which does not

require supervision or in-depth understanding of the text. We conducted an empirical study to assess the relevance of the criterion on the CNN/DM subset. This empirical study showed that around 20% of entities in the generated summaries were out of the document. In addition, out-of-document entities are mostly hallucinated, while in-document entities are most of the types correctly used in the summary. As the in-document entities contain variation, we proposed a heuristic to detect them. Because out-of-document entities are most of the time hallucination and the NEHR criterion, we proposed two approaches to reduce entity hallucination. The first one was based on selecting a summary that minimizes the NEHR criterion among a set of generated summaries obtained using the sampling method. The second consists in forcing the model during training by modifying the loss function. We found that selecting summaries based on NEHR was a promising approach, as the resulting summaries had more consistent entities in relation to the source document. Regarding loss modification, the different proposed losses give undesired results. We discussed the cause of these results and hypothesized that the problem could be because it is a pre-trained model or because the NEHR criterion used in the loss function does not have the same distribution as the cross entropy loss function (the usual function to train a summarization model).

Then, we focus on dialog summarization, particularly task-oriented dialog summarization, in Chapter 4. Rather than text summarization, task-oriented conversations involve exchanges between speakers for a specific purpose. We leveraged this unique aspect of task-oriented conversations to enhance the faithfulness of our summary regarding the task information. Our study was based on the DECODA dataset, which provides a customer service conversation between an agent and a customer of the public transportation service of Paris. Using the semantic information about the task, such as named entities or call type, we proposed two approaches to integrate such information to perform SLU summarization. First during training and second during the generation phase. For the first one, we proposed first to concatenate the call type to the input conversation to help the model generate faithful summaries regarding named entities. Then, we proposed to use three systems based on multitask generation. Indeed, we forced the model to generate the semantic task information or the set of information before generating the summary. During the generation phase, we introduced two selection criteria based on the semantic information about the task to select the summary that minimizes errors in this information. To evaluate the different proposals, we introduce four new criteria based on the semantic task information to evaluate the summaries in terms of this information. We believe that guaranteeing the semantics fidelity of certain task-related information will provide a more faithful summary. As a result, we found that minimizing the different criteria allows a summary more faithful regarding the semantic criteria. More human evaluations need to be performed to confirm our findings.

We also created a new version of the dataset DECODA with more data and a homogeneous summary. To do so, we used LLMs-based models such as ChatGPT to normalize and increase the data in the dataset by generating a summary. As manual transcription for dialog summarization is costly, we decided to explore the impact of ASR system such as WhisperX on the text summarization system. We found that using automatic transcription is a viable alternative to manual transcription. In addition, using task-information metric to select summary allow to increase the score of semantic metric such as scores on named entities.

Our findings confirmed that incorporating task-related semantic information leads to more semantically summaries regarding the source document, mainly when the training data covers a range of literary styles. Furthermore, we discovered that utilizing augmented data and transcription systems does not significantly affect the quality of the generated summaries when the word error rate of the transcription is low. By integrating task-related semantic information, we were able to compensate for the loss of semantic score caused by using automatic transcription of LLM-generated summaries. However, human evaluation and significance score need to be done to confirm our result.

Our research question entailed determining the most effective method for preserving faithfulness to the source material while generating a summary. While summary selection has proven fruitful, minimizing loss has not yielded the expected results. Nonetheless, selecting the appropriate criteria for summarization remains essential. We have identified several avenues that require further exploration to pursue this goal.

Future works

The findings of this thesis offer meaningful insights into avenues for future research.

Improving the training process of generative models through dedicated loss functions: In chapter 3, we proposed different loss function and found that the results was not what expected as the cross entropy and the NEHR have not the same distribution. We decided to optimize the cross entropy loss function using a RL system based on the NEHR criteria or any type of approach to estimate the faithfulness of generated summary such as QA-based approach. Furthermore, we can also employ RLHF to optimize our training using the human feedback on the faithfulness of the summary generated. We believe that optimizing a loss function using RL, will help to bridge this imbalance.

Modifying the decoding process that generates words from a given model and hallucination assessment: [CCK22] proposed a method to guide the text generation using MCTS based on the re-ranking of diverse sequences using the discriminator scores. As they obtained promising results, we want to explore the method by changing the discriminator scores. We believe that using a criterion based on our NEHR criterion can be used to obtain faithful summary regard to named entities.

Exploration of the use of other information in task-oriented dialog summarization to increase faithfulness: In our study, we explored the integration of call type and named entities to the summarization process. As further work, we want to explore how other type of information such as frame semantics (FrameNet), semantic units (SCU) can help improving the summary faithfulness. In dialog summarization, as each conversation have a structure, exploring the impact of the dialog structure to produce the summarization is also a path to explore.

Exploration of the use of task information in End-to-end method for text summarization: As our task-oriented summarization was based on Pipeline method (used of ASR system followed by automatic text summarization), we want to see the impact of our study using end-to-end method for dialog summarization. In addition to automatic transcription, Whisper has been trained to translate into English. Thus, this particular ability could have him to do summarization of raw audio. Therefore, It would be interesting to re-examine all the experiences of the thesis in this context.

Study the quality-computation compromise with various size of LLMs: We could repeat all the experiments of this thesis with the latest LLMs. We believe that the results should be similar, even if the quality of the summaries produced should obviously increase in view of LLMs' ability to generate fluent and more coherent text. In addition, we want to explore the latest breed of open LLMs such as LLama or Mistral [Tou+23b; Tou+23a; Jia+23] to normalize and increase the dataset. In the task-oriented dialog study, we used ChatGPT to normalize and increase the dataset. This can be a problem when confidential data is involved. Therefore, used open source LLMs can be a good alternative.

Bibliography

[Aka22] Eunice Akani. “Abstraction ou hallucination ? État des lieux et évaluation du risque pour les modèles de génération de résumés automatiques de type séquence-à-séquence”. In: *Journées Jointes des Groupements de Recherche Linguistique Informatique, Formelle et de Terrain (LIFT) et Traitement Automatique des Langues (TAL)*. Ed. by Leonor Becerra et al. Marseille, France: CNRS, 2022, pp. 9–12. URL: https://hal.science/hal-03846830 (cit. on p. 84).

[Aka23] Eunice Akani. “Étude de la fidélité des entités dans les résumés par abstraction”. French. In: *Actes de CORIA-TALN 2023. Actes des 16e Rencontres Jeunes Chercheurs en RI (RJCRI) et 25e Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL)*. Ed. by Marie Candito, Thomas Gerald, and José G Moreno. Paris, France: ATALA, June 2023, pp. 21–36. URL: https://aclanthology.org/2023.jeptalnrecital-rjc.2 (cit. on p. 111).

[AFB22] Eunice Akani, Benoit Favre, and Frederic Bechet. “Abstraction ou hallucination ? État des lieux et évaluation du risque pour les modèles de génération de résumés automatiques de type séquence-à-séquence (Abstraction or Hallucination ? Status and Risk assessment for sequence-to-sequence Automatic)”. French. In: *Actes de la 29e Conférence sur le Traitement Automatique des Langues Naturelles. Volume 1 : conférence principale*. Ed. by Yannick Estève et al. Avignon, France: ATALA, June 2022, pp. 2–11. URL: https://aclanthology.org/2022.jeptalnrecital-taln.1 (cit. on p. 84).

[Aka+23] Eunice Akani et al. “Reducing named entity hallucination risk to ensure faithful summary generation”. In: *Proceedings of the 16th International Natural Language Generation Conference*. Ed. by C. Maria Keet, Hung-Yi Lee, and Sina Zarrieß. Prague, Czechia: Association for Computational Linguistics, Sept. 2023, pp. 437–442. DOI: 10.18653/v1/2023.inlg-main.33. URL: https://aclanthology.org/2023.inlg-main.33 (cit. on p. 111).

[Aka+24a] Eunice Akani et al. “Approche multitâche pour l’amélioration de la fiabilité des systèmes de résumé automatique de conversation”. French. In: *Actes de la 31ème Conférence sur le Traitement Automatique des Langues Naturelles, volume 1 : articles longs et prises de position*. Ed. by Mathieu Balaguer et al. Toulouse, France: ATALA and AFPC, July 2024, pp. 338–351.

URL: https://aclanthology.org/2024.jeptalnrecital-taln.23 (cit. on p. 148).

[Aka+24b] Eunice Akani et al. "Unified Framework for Spoken Language Understanding and Summarization in Task-Based Human Dialog processing". In: *INTERSPEECH 2024*. 2024 (cit. on p. 148).

[ABV18] Alan Akbik, Duncan Blythe, and Roland Vollgraf. "Contextual String Embeddings for Sequence Labeling". In: *Proceedings of the 27th International Conference on Computational Linguistics*. Ed. by Emily M. Bender, Leon Derczynski, and Pierre Isabelle. Santa Fe, New Mexico, USA: Association for Computational Linguistics, Aug. 2018, pp. 1638–1649. URL: https://aclanthology.org/C18-1139 (cit. on p. 81).

[Akb+19] Alan Akbik et al. "FLAIR: An easy-to-use framework for state-of-the-art NLP". In: *NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*. 2019, pp. 54–59 (cit. on pp. 90, 107).

[BCB16] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. *Neural Machine Translation by Jointly Learning to Align and Translate*. 2016. arXiv: 1409.0473 [cs.CL] (cit. on p. 42).

[Bai+23] Max Bain et al. "WhisperX: Time-Accurate Speech Transcription of Long-Form Audio". In: *INTERSPEECH 2023* (2023) (cit. on p. 134).

[BL05] Satanjeev Banerjee and Alon Lavie. "METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments". In: *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*. Ann Arbor, Michigan: Association for Computational Linguistics, June 2005, pp. 65–72. URL: https://aclanthology.org/W05-0909 (cit. on pp. 58, 60, 70).

[BMW00] Michele Banko, Vibhu O. Mittal, and Michael J. Witbrock. "Headline Generation Based on Statistical Translation". In: *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*. Hong Kong: Association for Computational Linguistics, Oct. 2000, pp. 318–325. DOI: 10.3115/1075218.1075259. URL: https://aclanthology.org/P00-1041 (cit. on p. 40).

[Bec+12] Frederic Bechet et al. "DECODA: a call-centre human-human spoken conversation corpus". In: *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*. Istanbul, Turkey: European Language Resources Association (ELRA), May 2012, pp. 1343–1347. URL: http://www.lrec-conf.org/proceedings/lrec2012/pdf/684_Paper.pdf (cit. on pp. 53, 114, 116, 138).

[BC20] Rishi Bommasani and Claire Cardie. "Intrinsic Evaluation of Summarization Datasets". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 8075–8096. DOI: 10.18653/v1/2020.emnlp-main.649. URL: https://aclanthology.org/2020.emnlp-main.649 (cit. on p. 39).

[Bro+20] Tom B. Brown et al. *Language Models are Few-Shot Learners*. 2020. arXiv: 2005.14165 [cs.CL] (cit. on p. 135).

[Bud+18] Paweł Budzianowski et al. "MultiWOZ - A Large-Scale Multi-Domain Wizard-of-Oz Dataset for Task-Oriented Dialogue Modelling". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 5016–5026. DOI: 10.18653/v1/D18-1547. URL: https://aclanthology.org/D18-1547 (cit. on pp. 53, 116).

[CW21] Shuyang Cao and Lu Wang. "CLIFF: Contrastive Learning for Improving Faithfulness and Factuality in Abstractive Summarization". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 6633–6649. DOI: 10.18653/v1/2021.emnlp-main.532. URL: https://aclanthology.org/2021.emnlp-main.532 (cit. on p. 67).

[Cao+18] Ziqiang Cao et al. "Faithful to the Original: Fact Aware Neural Abstractive Summarization". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 32.1 (Apr. 2018). DOI: 10.1609/aaai.v32i1.11912. URL: https://ojs.aaai.org/index.php/AAAI/article/view/11912 (cit. on pp. 60, 66).

[Car+05] Jean Carletta et al. "The AMI Meeting Corpus: A Pre-announcement". In: *Machine Learning for Multimodal Interaction*. 2005. URL: https://api.semanticscholar.org/CorpusID:6118869 (cit. on pp. 52, 55, 116).

[CCK22] Antoine Chaffin, Vincent Claveau, and Ewa Kijak. "PPL-MCTS: Constrained Textual Generation Through Discriminator-Guided MCTS Decoding". In: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Ed. by Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz. Seattle, United States: Association for Computational Linguistics, July 2022, pp. 2953–2967. DOI: 10.18653/v1/2022.naacl-main.215. URL: https://aclanthology.org/2022.naacl-main.215 (cit. on pp. 14, 110, 151).

[CGO10] Éric Charton, Michel Gagnon, and Benoit Ozell. "Poly-co: An Unsupervised Co-reference Detection System". In: *Proceedings of the 6th International Natural Language Generation Conference*. Ed. by John Kelleher, Brian Mac Namee, and Ielka van der Sluis. Association for Computational

Linguistics, July 2010. URL: https://aclanthology.org/W10-4228 (cit. on p. 94).

[Che+20] Anthony Chen et al. "MOCHA: A Dataset for Training and Evaluating Generative Reading Comprehension Metrics". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 6521–6532. DOI: 10.18653/v1/2020.emnlp-main.528. URL: https://aclanthology.org/2020.emnlp-main.528 (cit. on p. 64).

[CY20] Jiaao Chen and Diyi Yang. "Multi-View Sequence-to-Sequence Models with Conversational Structure for Abstractive Dialogue Summarization". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Ed. by Bonnie Webber et al. Online: Association for Computational Linguistics, Nov. 2020, pp. 4106–4118. DOI: 10.18653/v1/2020.emnlp-main.336. URL: https://aclanthology.org/2020.emnlp-main.336 (cit. on p. 56).

[Che+21a] Sihao Chen et al. "Improving Faithfulness in Abstractive Summarization with Contrast Candidate Generation and Selection". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, June 2021, pp. 5935–5941. DOI: 10.18653/v1/2021.naacl-main.475. URL: https://aclanthology.org/2021.naacl-main.475 (cit. on pp. 67, 87).

[CB18] Yen-Chun Chen and Mohit Bansal. "Fast Abstractive Summarization with Reinforce-Selected Sentence Rewriting". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 675–686. DOI: 10.18653/v1/P18-1063. URL: https://aclanthology.org/P18-1063 (cit. on p. 46).

[Che+21b] Yulong Chen et al. "DialogSum: A Real-Life Scenario Dialogue Summarization Dataset". In: *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Online: Association for Computational Linguistics, Aug. 2021, pp. 5062–5074. DOI: 10.18653/v1/2021.findings-acl.449. URL: https://aclanthology.org/2021.findings-acl.449 (cit. on p. 53).

[CAR16] Sumit Chopra, Michael Auli, and Alexander M. Rush. "Abstractive Sentence Summarization with Attentive Recurrent Neural Networks". In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, June 2016, pp. 93–98. DOI: 10.18653/v1/N16-1012. URL: https://aclanthology.org/N16-1012 (cit. on pp. 42, 44, 60).

[Chu+14] Junyoung Chung et al. *Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling*. 2014. arXiv: 1412.3555 [cs.NE] (cit. on pp. 42, 43).

[Cla+20] Kevin Clark et al. *ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators*. 2020. arXiv: 2003.10555 [cs.CL] (cit. on p. 64).

[Coh+18] Arman Cohan et al. "A Discourse-Aware Attention Model for Abstractive Summarization of Long Documents". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pp. 615–621. DOI: 10.18653/v1/N18-2097. URL: https://aclanthology.org/N18-2097 (cit. on p. 38).

[Coh60] Jacob Cohen. "A Coefficient of Agreement for Nominal Scales". In: *Educational and Psychological Measurement* 20.1 (1960), pp. 37–46. DOI: 10.1177/001316446002000104. eprint: https://doi.org/10.1177/001316446002000104. URL: https://doi.org/10.1177/001316446002000104 (cit. on p. 93).

[Cui+20] Leyang Cui et al. "MuTual: A Dataset for Multi-Turn Dialogue Reasoning". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Ed. by Dan Jurafsky et al. Online: Association for Computational Linguistics, July 2020, pp. 1406–1416. DOI: 10.18653/v1/2020.acl-main.130. URL: https://aclanthology.org/2020.acl-main.130 (cit. on p. 53).

[Dan05] Hoa Trang Dang. "Overview of DUC 2005". In: *In Proceedings of the Document Understanding Conf. Wksp. 2005 (DUC 2005) at the Human Language Technology Conf./Conf. on Empirical Methods in Natural Language Processing (HLT/EMNLP*. 2005 (cit. on p. 71).

[Dan06] Hoa Trang Dang. "DUC 2005: Evaluation of Question-Focused Summarization Systems". In: *Proceedings of the Workshop on Task-Focused Summarization and Question Answering*. Sydney, Australia: Association for Computational Linguistics, July 2006, pp. 48–55. URL: https://aclanthology.org/W06-0707 (cit. on p. 37).

[DGL18] Dorottya Demszky, Kelvin Guu, and Percy Liang. *Transforming Question Answering Datasets Into Natural Language Inference Datasets*. 2018. arXiv: 1809.02922 [cs.CL] (cit. on p. 63).

[Dev+04] Laurence Devillers et al. "The French MEDIA/EVALDA Project: the Evaluation of the Understanding Capability of Spoken Language Dialogue Systems". In: *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*. Ed. by Maria Teresa Lino et al. Lisbon, Portugal: European Language Resources Association (ELRA), May

2004. URL: http://www.lrec-conf.org/proceedings/lrec2004/pdf/356.pdf (cit. on p. 120).

[Dev+19] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: https://aclanthology.org/N19-1423 (cit. on p. 48).

[Don+20] Yue Dong et al. "Multi-Fact Correction in Abstractive Text Summarization". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 9320–9331. DOI: 10.18653/v1/2020.emnlp-main.749. URL: https://aclanthology.org/2020.emnlp-main.749 (cit. on p. 61).

[DHD20] Esin Durmus, He He, and Mona Diab. "FEQA: A Question Answering Evaluation Framework for Faithfulness Assessment in Abstractive Summarization". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 5055–5070. DOI: 10.18653/v1/2020.acl-main.454. URL: https://aclanthology.org/2020.acl-main.454 (cit. on pp. 63, 83).

[Dye02] Dick Dye. "Abstraction". In: *Encyclopedia of Software Engineering*. John Wiley & Sons, Ltd, 2002. ISBN: 9780471028956. DOI: https://doi.org/10.1002/0471028959.sof002. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/0471028959.sof002. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/0471028959.sof002 (cit. on p. 73).

[Elm90] Jeffrey L. Elman. "Finding structure in time". In: *Cognitive Science* 14.2 (1990), pp. 179–211. ISSN: 0364-0213. DOI: https://doi.org/10.1016/0364-0213(90)90002-E. URL: https://www.sciencedirect.com/science/article/pii/036402139090002E (cit. on p. 42).

[Fab+19] Alexander Fabbri et al. "Multi-News: A Large-Scale Multi-Document Summarization Dataset and Abstractive Hierarchical Model". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 1074–1084. DOI: 10.18653/v1/P19-1102. URL: https://aclanthology.org/P19-1102 (cit. on p. 37).

[Fab+22] Alexander Fabbri et al. "QAFactEval: Improved QA-Based Factual Consistency Evaluation for Summarization". In: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Ed. by Marine Carpuat,

Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz. Seattle, United States: Association for Computational Linguistics, July 2022, pp. 2587–2601. DOI: 10.18653/v1/2022.naacl-main.187. URL: https://aclanthology.org/2022.naacl-main.187 (cit. on p. 63).

[Fab+21] Alexander R. Fabbri et al. *SummEval: Re-evaluating Summarization Evaluation*. 2021. arXiv: 2007.12626 [cs.CL] (cit. on p. 66).

[Fal+19] Tobias Falke et al. "Ranking Generated Summaries by Correctness: An Interesting but Challenging Application for Natural Language Inference". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 2214–2220. DOI: 10.18653/v1/P19-1213. URL: https://aclanthology.org/P19-1213 (cit. on pp. 60, 62, 66, 67).

[FLD18] Angela Fan, Mike Lewis, and Yann Dauphin. "Hierarchical Neural Story Generation". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Iryna Gurevych and Yusuke Miyao. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 889–898. DOI: 10.18653/v1/P18-1082. URL: https://aclanthology.org/P18-1082 (cit. on p. 89).

[Fav+15] Benoit Favre et al. "Call Centre Conversation Summarization: A Pilot Task at Multiling 2015". In: *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Ed. by Alexander Koller et al. Prague, Czech Republic: Association for Computational Linguistics, Sept. 2015, pp. 232–236. DOI: 10.18653/v1/W15-4633. URL: https://aclanthology.org/W15-4633 (cit. on p. 57).

[Fen+21] Xiachong Feng et al. "Language Model as an Annotator: Exploring DialoGPT for Dialogue Summarization". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Ed. by Chengqing Zong et al. Online: Association for Computational Linguistics, Aug. 2021, pp. 1479–1491. DOI: 10.18653/v1/2021.acl-long.117. URL: https://aclanthology.org/2021.acl-long.117 (cit. on p. 56).

[Fil10] Katja Filippova. "Multi-Sentence Compression: Finding Shortest Paths in Word Graphs". In: *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*. Ed. by Chu-Ren Huang and Dan Jurafsky. Beijing, China: Coling 2010 Organizing Committee, Aug. 2010, pp. 322–330. URL: https://aclanthology.org/C10-1037 (cit. on p. 55).

[Gab+21] Saadia Gabriel et al. "GO FIGURE: A Meta Evaluation of Factuality in Summarization". In: *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Online: Association for Computational Linguistics, Aug. 2021, pp. 478–487. DOI: 10.18653/v1/2021.findings-acl.42.

URL: https://aclanthology.org/2021.findings-acl.42 (cit. on p. 63).

[GDR18] Sebastian Gehrmann, Yuntian Deng, and Alexander Rush. "Bottom-Up Abstractive Summarization". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 4098–4109. DOI: 10.18653/v1/D18-1443. URL: https://aclanthology.org/D18-1443 (cit. on p. 46).

[Gli+19] Bogdan Gliwa et al. "SAMSum Corpus: A Human-annotated Dialogue Dataset for Abstractive Summarization". In: *Proceedings of the 2nd Workshop on New Frontiers in Summarization*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 70–79. DOI: 10.18653/v1/D19-5409. URL: https://aclanthology.org/D19-5409 (cit. on pp. 52, 56, 67, 68).

[GC18] Chih-Wen Goo and Yun-Nung Chen. *Abstractive Dialogue Summarization with Sentence-Gated Modeling Optimized by Dialogue Acts*. 2018. arXiv: 1809.05715 [cs.CL] (cit. on p. 55).

[Gra+03] David Graff et al. "English gigaword". In: *Linguistic Data Consortium, Philadelphia* 4.1 (2003), p. 34 (cit. on p. 37).

[GNA18] Max Grusky, Mor Naaman, and Yoav Artzi. "Newsroom: A Dataset of 1.3 Million Summaries with Diverse Extractive Strategies". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pp. 708–719. DOI: 10.18653/v1/N18-1065. URL: https://aclanthology.org/N18-1065 (cit. on pp. 37, 39).

[HO04] Donna Harman and Paul Over. "The Effects of Human Variation in DUC Summarization Evaluation". In: *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, July 2004, pp. 10–17. URL: https://aclanthology.org/W04-1003 (cit. on p. 37).

[Has+21] Tahmid Hasan et al. "XL-Sum: Large-Scale Multilingual Abstractive Summarization for 44 Languages". In: *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Online: Association for Computational Linguistics, Aug. 2021, pp. 4693–4703. DOI: 10.18653/v1/2021.findings-acl.413. URL: https://aclanthology.org/2021.findings-acl.413 (cit. on pp. 38, 39, 76).

[HGD90] Charles T. Hemphill, John J. Godfrey, and George R. Doddington. "The ATIS Spoken Language Systems Pilot Corpus". In: *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27,1990*. 1990. URL: https://aclanthology.org/H90-1021 (cit. on p. 120).

[Her+15] Karl Moritz Hermann et al. "Teaching Machines to Read and Comprehend". In: *NIPS*. 2015, pp. 1693–1701. URL: http://papers.nips.cc/paper/5945-teaching-machines-to-read-and-comprehend (cit. on pp. 37, 88, 91, 98).

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory". In: *Neural Computation* 9.8 (Nov. 1997), pp. 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. eprint: https://direct.mit.edu/neco/article-pdf/9/8/1735/813796/neco.1997.9.8.1735.pdf. URL: https://doi.org/10.1162/neco.1997.9.8.1735 (cit. on pp. 42, 43).

[Hol+19] Ari Holtzman et al. *The Curious Case of Neural Text Degeneration*. 2019. DOI: 10.48550/ARXIV.1904.09751. URL: https://arxiv.org/abs/1904.09751 (cit. on p. 89).

[Hua+20] Dandan Huang et al. "What Have We Achieved on Text Summarization?" In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Ed. by Bonnie Webber et al. Online: Association for Computational Linguistics, Nov. 2020, pp. 446–469. DOI: 10.18653/v1/2020.emnlp-main.33. URL: https://aclanthology.org/2020.emnlp-main.33 (cit. on p. 66).

[Hua+23] Kung-Hsiang Huang et al. "SWING: Balancing Coverage and Faithfulness for Dialogue Summarization". In: *Findings of the Association for Computational Linguistics: EACL 2023*. Ed. by Andreas Vlachos and Isabelle Augenstein. Dubrovnik, Croatia: Association for Computational Linguistics, May 2023, pp. 512–525. DOI: 10.18653/v1/2023.findings-eacl.37. URL: https://aclanthology.org/2023.findings-eacl.37 (cit. on p. 69).

[HCL22] Sicong Huang, Asli Celikyilmaz, and Haoran Li. *ED-FAITH: Evaluating Dialogue Summarization on Faithfulness*. 2022. arXiv: 2211.08464 [cs.CL] (cit. on p. 68).

[Jan+03] A. Janin et al. "The ICSI Meeting Corpus". In: *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03)*. Vol. 1. 2003, pp. I–I. DOI: 10.1109/ICASSP.2003.1198793 (cit. on pp. 52, 55, 116).

[Jea+15] Sébastien Jean et al. "On Using Very Large Target Vocabulary for Neural Machine Translation". In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Beijing, China: Association for Computational Linguistics, July 2015, pp. 1–10. DOI: 10.3115/v1/P15-1001. URL: https://aclanthology.org/P15-1001 (cit. on p. 42).

[Jia+23] Albert Q. Jiang et al. *Mistral 7B*. 2023. arXiv: 2310.06825 [cs.CL] (cit. on pp. 135, 152).

[KTV21] Moussa Kamal Eddine, Antoine Tixier, and Michalis Vazirgiannis. "BARThez: a Skilled Pretrained French Sequence-to-Sequence Model". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 9369–9390. DOI: 10.18653/v1/2021.emnlp-main.740. URL: https://aclanthology.org/2021.emnlp-main.740 (cit. on pp. 74, 76, 77, 126, 127, 139, 140, 143).

[KZS21] Misha Khalman, Yao Zhao, and Mohammad Saleh. "ForumSum: A Multi-Speaker Conversation Summarization Dataset". In: *Findings of the Association for Computational Linguistics: EMNLP 2021*. Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 4592–4599. DOI: 10.18653/v1/2021.findings-emnlp.391. URL: https://aclanthology.org/2021.findings-emnlp.391 (cit. on p. 53).

[KKK19] Byeongchang Kim, Hyunwoo Kim, and Gunhee Kim. "Abstractive Summarization of Reddit Posts with Multi-level Memory Networks". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 2519–2531. DOI: 10.18653/v1/N19-1260. URL: https://aclanthology.org/N19-1260 (cit. on p. 38).

[KKK18] Seonhoon Kim, Inho Kang, and Nojun Kwak. *Semantic Sentence Matching with Densely-connected Recurrent and Co-attentive Information*. 2018. arXiv: 1805.11360 [cs.CL] (cit. on p. 62).

[KY04] Bryan Klimt and Yiming Yang. "The Enron Corpus: A New Dataset for Email Classification Research". In: *Machine Learning: ECML 2004*. Ed. by Jean-François Boulicaut et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 217–226. ISBN: 978-3-540-30115-8 (cit. on p. 38).

[KM02] Kevin Knight and Daniel Marcu. "Summarization beyond sentence extraction: A probabilistic approach to sentence compression". In: *Artificial Intelligence* 139.1 (2002), pp. 91–107. ISSN: 0004-3702. DOI: https://doi.org/10.1016/S0004-3702(02)00222-9. URL: https://www.sciencedirect.com/science/article/pii/S0004370202002229 (cit. on p. 40).

[Koa+20] Jia Jin Koay et al. "How Domain Terminology Affects Meeting Summarization Performance". In: *Proceedings of the 28th International Conference on Computational Linguistics*. Barcelona, Spain (Online): International Committee on Computational Linguistics, Dec. 2020, pp. 5689–5695. DOI:

10.18653/v1/2020.coling-main.499. URL: https://aclanthology.org/2020.coling-main.499 (cit. on p. 55).

[KW18] Mahnaz Koupaee and William Yang Wang. *WikiHow: A Large Scale Text Summarization Dataset*. 2018. arXiv: 1810.09305 [cs.CL] (cit. on p. 38).

[Kry+20] Wojciech Kryscinski et al. "Evaluating the Factual Consistency of Abstractive Text Summarization". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 9332–9346. DOI: 10.18653/v1/2020.emnlp-main.750. URL: https://aclanthology.org/2020.emnlp-main.750 (cit. on pp. 64, 66).

[KL51] S. Kullback and R. A. Leibler. "On Information and Sufficiency". In: *The Annals of Mathematical Statistics* 22.1 (1951). Publisher: Institute of Mathematical Statistics, pp. 79–86. ISSN: 0003-4851. URL: https://www.jstor.org/stable/2236703 (cit. on p. 122).

[Lab+22] Philippe Laban et al. "SummaC: Re-Visiting NLI-based Models for Inconsistency Detection in Summarization". In: *Transactions of the Association for Computational Linguistics* 10 (2022). Ed. by Brian Roark and Ani Nenkova, pp. 163–177. DOI: 10.1162/tacl_a_00453. URL: https://aclanthology.org/2022.tacl-1.10 (cit. on pp. 62, 63, 65, 66).

[Lee+18] Jihwan Lee et al. "Coupled representation learning for domains, intents and slots in spoken language understanding". In: *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE. 2018, pp. 714–719 (cit. on p. 120).

[Lew+20] Mike Lewis et al. "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 7871–7880. DOI: 10.18653/v1/2020.acl-main.703. URL: https://aclanthology.org/2020.acl-main.703 (cit. on pp. 48, 49, 55, 56, 64, 69, 76, 88, 98, 107, 108, 127).

[Li+18] Haoran Li et al. "Ensure the Correctness of the Summary: Incorporate Entailment Knowledge into Abstractive Sentence Summarization". In: *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, Aug. 2018, pp. 1430–1441. URL: https://aclanthology.org/C18-1121 (cit. on p. 66).

[Li+17] Yanran Li et al. "DailyDialog: A Manually Labelled Multi-turn Dialogue Dataset". In: *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Ed. by Greg Kondrak and Taro Watanabe. Taipei, Taiwan: Asian Federation of Natural Language Processing, Nov. 2017, pp. 986–995. URL: https://aclanthology.org/I17-1099 (cit. on p. 53).

[Lin04] Chin-Yew Lin. "ROUGE: A Package for Automatic Evaluation of Summaries". In: *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, July 2004, pp. 74–81. URL: https://aclanthology.org/W04-1013 (cit. on pp. 46, 58, 60, 68, 70, 88, 90, 98, 108, 124, 130, 140, 144).

[Lin+21] Haitao Lin et al. "CSDS: A Fine-Grained Chinese Dataset for Customer Service Dialogue Summarization". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 4436–4451. DOI: 10.18653/v1/2021.emnlp-main.365. URL: https://aclanthology.org/2021.emnlp-main.365 (cit. on pp. 53, 54, 116).

[Liu+19a] Chunyi Liu et al. "Automatic Dialogue Summary Generation for Customer Service". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. KDD '19. Anchorage, AK, USA: Association for Computing Machinery, 2019, pp. 1957–1965. ISBN: 9781450362016. DOI: 10.1145/3292500.3330683. URL: https://doi.org/10.1145/3292500.3330683 (cit. on p. 57).

[LL19] Yang Liu and Mirella Lapata. "Text Summarization with Pretrained Encoders". In: *arXiv:1908.08345 [cs]* (Sept. 2019). arXiv: 1908.08345. URL: http://arxiv.org/abs/1908.08345 (visited on 03/11/2021) (cit. on p. 47).

[Liu+19b] Yinhan Liu et al. *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. 2019. arXiv: 1907.11692 [cs.CL] (cit. on p. 76).

[Liu+19c] Yinhan Liu et al. "RoBERTa: A Robustly Optimized BERT Pretraining Approach". In: *CoRR* abs/1907.11692 (2019). arXiv: 1907.11692. URL: http://arxiv.org/abs/1907.11692 (cit. on p. 127).

[LC21] Zhengyuan Liu and Nancy Chen. "Controllable Neural Dialogue Summarization with Personal Named Entity Planning". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 92–106. DOI: 10.18653/v1/2021.emnlp-main.8. URL: https://aclanthology.org/2021.emnlp-main.8 (cit. on pp. 56, 122).

[Mar+20] Louis Martin et al. "CamemBERT: a Tasty French Language Model". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 7203–7219. DOI: 10.18653/v1/2020.acl-main.645. URL: https://aclanthology.org/2020.acl-main.645 (cit. on pp. 75, 125, 127, 128, 142).

[Mat+23] Kohei Matsuura et al. “Leveraging large text corpora for end-to-end speech summarization”. In: *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2023, pp. 1–5 (cit. on p. 115).

[May+20] Joshua Maynez et al. “On Faithfulness and Factuality in Abstractive Summarization”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 1906–1919. DOI: 10.18653/v1/2020.acl-main.173. URL: https://aclanthology.org/2020.acl-main.173 (cit. on pp. 30, 60–62, 66, 70, 73, 97).

[Mis+21] Anshuman Mishra et al. “Looking Beyond Sentence-Level Natural Language Inference for Question Answering and Text Summarization”. In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, June 2021, pp. 1322–1336. DOI: 10.18653/v1/2021.naacl-main.104. URL: https://aclanthology.org/2021.naacl-main.104 (cit. on p. 62).

[Nak18] Hiroki Nakayama. *seqeval: A Python framework for sequence labeling evaluation*. Software available from https://github.com/chakki-works/seqeval. 2018. URL: https://github.com/chakki-works/seqeval (cit. on p. 128).

[Nal+16] Ramesh Nallapati et al. “Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond”. In: *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 280–290. DOI: 10.18653/v1/K16-1028. URL: https://aclanthology.org/K16-1028 (cit. on pp. 37, 38, 42–46, 48–50, 65, 67).

[Nan+21a] Feng Nan et al. “Entity-level Factual Consistency of Abstractive Text Summarization”. In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Online: Association for Computational Linguistics, Apr. 2021, pp. 2727–2733. DOI: 10.18653/v1/2021.eacl-main.235. URL: https://aclanthology.org/2021.eacl-main.235 (cit. on pp. 64, 67, 87, 125).

[Nan+21b] Feng Nan et al. “Improving Factual Consistency of Abstractive Summarization via Question Answering”. In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, Aug. 2021, pp. 6881–6894. DOI: 10.18653/v1/2021.acl-long.536. URL: https://aclanthology.org/2021.acl-long.536 (cit. on p. 63).

[NCL18] Shashi Narayan, Shay B. Cohen, and Mirella Lapata. "Don't Give Me the Details, Just the Summary! Topic-Aware Convolutional Neural Networks for Extreme Summarization". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 1797–1807. DOI: 10.18653/v1/D18-1206. URL: https://aclanthology.org/D18-1206 (cit. on pp. 37, 39, 48, 50, 62, 65, 67, 73, 88, 91, 98, 106, 108).

[Nar+21] Shashi Narayan et al. "Planning with Learned Entity Prompts for Abstractive Summarization". In: *Transactions of the Association for Computational Linguistics* 9 (2021), pp. 1475–1492. DOI: 10.1162/tacl_a_00438. URL: https://aclanthology.org/2021.tacl-1.88 (cit. on pp. 67, 121, 122).

[NPM07] Ani Nenkova, Rebecca Passonneau, and Kathleen Mckeown. "The Pyramid Method: Incorporating human content selection variation in summarization evaluation". English (US). In: *ACM Transactions on Speech and Language Processing* 4.2 (May 2007). ISSN: 1550-4875. DOI: 10.1145/1233912.1233913 (cit. on p. 59).

[Obe19] Bruno Oberle. "Détection automatique de chaınes de coréférence pour le français écrit: règles et ressources adaptées au repérage de phénomènes linguistiques spécifiques (Automatic coreference resolution for written French : rules and resources for specific linguistic phenomena)". French. In: *Actes de la Conférence sur le Traitement Automatique des Langues Naturelles (TALN) PFIA 2019. Volume III : RECITAL*. Ed. by Emmanuel Morin, Sophie Rosset, and Pierre Zweigenbaum. Toulouse, France: ATALA, July 2019, pp. 499–512. URL: https://aclanthology.org/2019.jeptalnrecital-recital.4 (cit. on p. 94).

[Ope+24] OpenAI et al. *GPT-4 Technical Report*. 2024. arXiv: 2303.08774 [cs.CL] (cit. on p. 135).

[Oya+14] Tatsuro Oya et al. "A Template-based Abstractive Meeting Summarization: Leveraging Summary and Source Text Relationships". In: *Proceedings of the 8th International Natural Language Generation Conference (INLG)*. Ed. by Margaret Mitchell et al. Philadelphia, Pennsylvania, U.S.A.: Association for Computational Linguistics, June 2014, pp. 45–53. DOI: 10.3115/v1/W14-4407. URL: https://aclanthology.org/W14-4407 (cit. on p. 41).

[PBT21] Artidoro Pagnoni, Vidhisha Balachandran, and Yulia Tsvetkov. "Understanding Factuality in Abstractive Summarization with FRANK: A Benchmark for Factuality Metrics". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, June 2021, pp. 4812–4829. DOI: 10.18653/v1/2021.naacl-

main.383. URL: https://aclanthology.org/2021.naacl-main.383 (cit. on pp. 61, 66, 71).

[Pap+02] Kishore Papineni et al. "Bleu: a Method for Automatic Evaluation of Machine Translation". In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, July 2002, pp. 311–318. DOI: 10.3115/1073083.1073135. URL: https://aclanthology.org/P02-1040 (cit. on p. 60).

[PXS18] Romain Paulus, Caiming Xiong, and Richard Socher. "A Deep Reinforced Model for Abstractive Summarization". In: *International Conference on Learning Representations*. 2018. URL: https://openreview.net/forum?id=HkAClQgA- (cit. on pp. 37, 45).

[Rad+22] Alec Radford et al. *Robust Speech Recognition via Large-Scale Weak Supervision*. 2022. arXiv: 2212.04356 [eess.AS] (cit. on p. 134).

[Raf+20] Colin Raffel et al. "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer". In: *Journal of Machine Learning Research* 21.140 (2020), pp. 1–67. URL: http://jmlr.org/papers/v21/20-074.html (cit. on pp. 49, 50, 69, 76).

[RdP13] Marta Recasens, Marie-Catherine de Marneffe, and Christopher Potts. "The Life and Death of Discourse Entities: Identifying Singleton Mentions". In: *North American Association for Computational Linguistics (NAACL)*. 2013 (cit. on p. 94).

[Ren+17] Steven J. Rennie et al. *Self-critical Sequence Training for Image Captioning*. 2017. arXiv: 1612.00563 [cs.LG] (cit. on p. 45).

[RCW15] Alexander M. Rush, Sumit Chopra, and Jason Weston. "A Neural Attention Model for Abstractive Sentence Summarization". In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, Sept. 2015, pp. 379–389. DOI: 10.18653/v1/D15-1044. URL: https://aclanthology.org/D15-1044 (cit. on pp. 37, 41, 42, 44, 60, 66).

[SL00] Horacio Saggion and Guy Lapalme. "Concept Identification and Presentation in the Context of Technical Text Summarization". In: *NAACL-ANLP 2000 Workshop: Automatic Summarization*. 2000. URL: https://aclanthology.org/W00-0401 (cit. on p. 41).

[San08] Evan Sandhaus. "The new york times annotated corpus". In: *Linguistic Data Consortium, Philadelphia* 6.12 (2008), e26752 (cit. on p. 37).

[SA20] Stefan Schweter and Alan Akbik. *FLERT: Document-Level Features for Named Entity Recognition*. 2020. arXiv: 2011.06993 [cs.CL] (cit. on pp. 90, 107).

[Sci+20] Thomas Scialom et al. “MLSUM: The Multilingual Summarization Corpus”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 8051–8067. DOI: 10.18653/v1/2020.emnlp-main.647. URL: https://aclanthology.org/2020.emnlp-main.647 (cit. on pp. 38, 75).

[Sci+21] Thomas Scialom et al. “QuestEval: Summarization Asks for Fact-based Evaluation”. In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 6594–6604. DOI: 10.18653/v1/2021.emnlp-main.529. URL: https://aclanthology.org/2021.emnlp-main.529 (cit. on p. 63).

[SLM17] Abigail See, Peter J. Liu, and Christopher D. Manning. “Get To The Point: Summarization with Pointer-Generator Networks”. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, July 2017, pp. 1073–1083. DOI: 10.18653/v1/P17-1099. URL: https://aclanthology.org/P17-1099 (cit. on pp. 44, 46, 55, 75, 76).

[Sha+18] Guokan Shang et al. “Unsupervised Abstractive Meeting Summarization with Multi-Sentence Compression and Budgeted Submodular Maximization”. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Iryna Gurevych and Yusuke Miyao. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 664–674. DOI: 10.18653/v1/P18-1062. URL: https://aclanthology.org/P18-1062 (cit. on p. 55).

[SLW19] Eva Sharma, Chen Li, and Lu Wang. “BIGPATENT: A Large-Scale Dataset for Abstractive and Coherent Summarization”. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 2204–2213. DOI: 10.18653/v1/P19-1212. URL: https://aclanthology.org/P19-1212 (cit. on p. 38).

[Sha+21] Roshan Sharma et al. “End-to-End Speech Summarization Using Restricted Self-Attention”. In: *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2021), pp. 8072–8076. URL: https://api.semanticscholar.org/CorpusID:249436553 (cit. on p. 115).

[Sha+22] Roshan Sharma et al. “End-to-end speech summarization using restricted self-attention”. In: *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2022, pp. 8072–8076 (cit. on p. 114).

[Sun+19] Kai Sun et al. “DREAM: A Challenge Data Set and Models for Dialogue-Based Reading Comprehension”. In: *Transactions of the Association for Computational Linguistics* 7 (2019). Ed. by Lillian Lee et al., pp. 217–231. DOI: 10.1162/tacl_a_00264. URL: https://aclanthology.org/Q19-1014 (cit. on p. 53).

[SVL14] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. *Sequence to Sequence Learning with Neural Networks*. 2014. arXiv: 1409.3215 [cs.CL] (cit. on p. 45).

[Tan+22] Xiangru Tang et al. “CONFIT: Toward Faithful Dialogue Summarization with Linguistically-Informed Contrastive Fine-tuning”. In: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Ed. by Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz. Seattle, United States: Association for Computational Linguistics, July 2022, pp. 5657–5668. DOI: 10.18653/v1/2022.naacl-main.415. URL: https://aclanthology.org/2022.naacl-main.415 (cit. on pp. 68, 69).

[Tou+23a] Hugo Touvron et al. *Llama 2: Open Foundation and Fine-Tuned Chat Models*. 2023. arXiv: 2307.09288 [cs.CL] (cit. on pp. 135, 152).

[Tou+23b] Hugo Touvron et al. *LLaMA: Open and Efficient Foundation Language Models*. 2023. arXiv: 2302.13971 [cs.CL] (cit. on pp. 135, 152).

[Tri17] Jérémy Trione. “Méthodes par abstraction et extraction pour le résumé de conversations orales issues de centres d’appels téléphoniques”. 2017AIXM0175. PhD thesis. 2017. URL: http://www.theses.fr/2017AIXM0175/document (cit. on p. 41).

[TFB16] Jérémy Trione, Benoit Favre, and Frederic Bechet. “Détection de concepts pertinents pour le résumé automatique de conversations par recombinaison de patrons (Relevant concepts detection for the automatic summary of conversations using patterns recombination)”. French. In: *Actes de la conférence conjointe JEP-TALN-RECITAL 2016. volume 2 : TALN (Articles longs)*. Ed. by Laurence Danlos and Thierry Hamon. Paris, France: AFCP - ATALA, July 2016, pp. 57–69. URL: https://aclanthology.org/2016.jeptalnrecital-long.5 (cit. on p. 57).

[Vas+17] Ashish Vaswani et al. “Attention is All you Need”. In: *ArXiv* abs/1706.03762 (2017). URL: http://arxiv.org/abs/1706.03762 (cit. on pp. 41, 46–49, 55, 108).

[WCL20] Alex Wang, Kyunghyun Cho, and Mike Lewis. “Asking and Answering Questions to Evaluate the Factual Consistency of Summaries”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 5008–5020. DOI: 10.18653/v1/2020.acl-main.450. URL: https://aclanthology.org/2020.acl-main.450 (cit. on p. 63).

[Wan+22] Bin Wang et al. “Analyzing and Evaluating Faithfulness in Dialogue Summarization”. In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 4897–4908. DOI: 10.18653/v1/2022.emnlp-main.325. URL: https://aclanthology.org/2022.emnlp-main.325 (cit. on p. 68).

[WZ89] Ronald J. Williams and David Zipser. “A Learning Algorithm for Continually Running Fully Recurrent Neural Networks”. In: *Neural Computation* 1.2 (1989), pp. 270–280. DOI: 10.1162/neco.1989.1.2.270 (cit. on p. 45).

[Wol+20] Thomas Wolf et al. “Transformers: State-of-the-Art Natural Language Processing”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Ed. by Qun Liu and David Schlangen. Online: Association for Computational Linguistics, Oct. 2020, pp. 38–45. DOI: 10.18653/v1/2020.emnlp-demos.6. URL: https://aclanthology.org/2020.emnlp-demos.6 (cit. on p. 88).

[Xu+16] Kelvin Xu et al. *Show, Attend and Tell: Neural Image Caption Generation with Visual Attention*. 2016. arXiv: 1502.03044 [cs.LG] (cit. on p. 44).

[Xue+21] Linting Xue et al. “mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer”. In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, June 2021, pp. 483–498. DOI: 10.18653/v1/2021.naacl-main.41. URL: https://aclanthology.org/2021.naacl-main.41 (cit. on p. 76).

[YNL21] Weizhe Yuan, Graham Neubig, and Pengfei Liu. “BARTScore: Evaluating Generated Text as Text Generation”. In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato et al. Vol. 34. Curran Associates, Inc., 2021, pp. 27263–27277 (cit. on pp. 64, 68).

[Zha+20a] Jingqing Zhang et al. “PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization”. In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by Hal Daumé III and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, July 2020, pp. 11328–11339. URL: https://proceedings.mlr.press/v119/zhang20ae.html (cit. on pp. 49, 50, 69).

[ZT19] Rui Zhang and Joel Tetreault. “This Email Could Save Your Life: Introducing the Task of Email Subject Line Generation”. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 446–456. DOI: 10.18653/v1/P19-1043. URL: https://aclanthology.org/P19-1043 (cit. on p. 38).

[Zha+20b] Xinyuan Zhang et al. "Unsupervised Abstractive Dialogue Summarization for Tete-a-Tetes". In: *AAAI Conference on Artificial Intelligence*. 2020. URL: https://api.semanticscholar.org/CorpusID:221702889 (cit. on p. 57).

[Zha+20c] Tianyi Zhang* et al. "BERTScore: Evaluating Text Generation with BERT". In: *International Conference on Learning Representations*. 2020. URL: https://openreview.net/forum?id=SkeHuCVFDr (cit. on pp. 59, 68, 70, 95, 124, 140, 144).

[ZXG20] Lulu Zhao, Weiran Xu, and Jun Guo. "Improving Abstractive Dialogue Summarization with Graph Structures and Topic Words". In: *Proceedings of the 28th International Conference on Computational Linguistics*. Barcelona, Spain (Online): International Committee on Computational Linguistics, Dec. 2020, pp. 437–449. DOI: 10.18653/v1/2020.coling-main.39. URL: https://aclanthology.org/2020.coling-main.39 (cit. on p. 56).

[Zha+21] Lulu Zhao et al. *TODSum: Task-Oriented Dialogue Summarization with State Tracking*. 2021. arXiv: 2110.12680 [cs.CL] (cit. on pp. 53, 116).

[ZCW20] Zheng Zhao, Shay B. Cohen, and Bonnie Webber. "Reducing Quantity Hallucinations in Abstractive Summarization". In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, Nov. 2020, pp. 2237–2249. DOI: 10.18653/v1/2020.findings-emnlp.203. URL: https://aclanthology.org/2020.findings-emnlp.203 (cit. on p. 67).

[Zho+21a] Ming Zhong et al. "QMSum: A New Benchmark for Query-based Multi-domain Meeting Summarization". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Ed. by Kristina Toutanova et al. Online: Association for Computational Linguistics, June 2021, pp. 5905–5921. DOI: 10.18653/v1/2021.naacl-main.472. URL: https://aclanthology.org/2021.naacl-main.472 (cit. on p. 55).

[Zho+21b] Chunting Zhou et al. "Detecting Hallucinated Content in Conditional Neural Sequence Generation". In: *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Online: Association for Computational Linguistics, Aug. 2021, pp. 1393–1404. DOI: 10.18653/v1/2021.findings-acl.120. URL: https://aclanthology.org/2021.findings-acl.120 (cit. on pp. 60, 64).

[ZPR22] Yongxin Zhou, François Portet, and Fabien Ringeval. "Effectiveness of French Language Models on Abstractive Dialogue Summarization Task". In: *Proceedings of the Thirteenth Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, June 2022, pp. 3571–3581. URL: https://aclanthology.org/2022.lrec-1.382 (cit. on p. 130).

[Zhu+20] Chenguang Zhu et al. “A Hierarchical Network for Abstractive Meeting Summarization with Cross-Domain Pretraining”. In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Ed. by Trevor Cohn, Yulan He, and Yang Liu. Online: Association for Computational Linguistics, Nov. 2020, pp. 194–203. DOI: 10.18653/v1/2020.findings-emnlp.19. URL: https://aclanthology.org/2020.findings-emnlp.19 (cit. on p. 55).

[Zhu+21] Chenguang Zhu et al. “Enhancing Factual Consistency of Abstractive Summarization”. In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, June 2021, pp. 718–733. DOI: 10.18653/v1/2021.naacl-main.58. URL: https://aclanthology.org/2021.naacl-main.58 (cit. on p. 67).

[Zou+21] Yicheng Zou et al. *Unsupervised Summarization for Chat Logs with Topic-Oriented Ranking and Context-Aware Auto-Encoders*. 2021. arXiv: 2012.07300 [cs.CL] (cit. on p. 57).