Machine Translation of User-Generated Contents : an Evaluation of Neural Translation Systems under Zero-shot Conditions

José Rosales Núñez

université
PARIS-SACLAY

Machine Translation of User-Generated Contents: an Evaluation of Neural Translation Systems under Zero-shot Conditions

Traduction Automatique de Contenus Générés par l'Utilisateur : une Évaluation des Systèmes de Traduction Neuronaux dans des Conditions Zero-shot

Thèse de doctorat de l'Université Paris-Saclay

École doctorale n°580 Sciences et Technologies de l'Information et de la Communication (STIC)
Spécialité de doctorat: Informatique

Graduate School : Informatique et sciences du numérique,
Référent : Faculté des sciences d'Orsay

Thèse préparée dans le **Laboratoire Interdisciplinaire des Sciences du Numérique** (Université Paris-Saclay, CNRS) et **Inria Paris**, sous la direction de **Guillaume WISNIEWSKI**, Maître de Conférences, et le co-encadrement de **Djamé SEDDAH**, Maître de Conférences.

Thèse soutenue à Paris, le 3 octobre 2023, par

José Carlos ROSALES NÚÑEZ

Composition du Jury

Membres du jury avec voix délibérative

Thierry POIBEAU Directeur de Recherche, CNRS (LATTICE)	Président & Rapporteur
Christophe CERISARA Chargé de Recherche (HDR), CNRS (LORIA)	Rapporteur & Examinateur
Joseph LE ROUX Maître de Conférences, Université Paris 13 Nord (LIPN)	Examinateur
Anne VILNAT Professeure, Université Paris-Saclay (LISN)	Examinatrice

Thèse de doctorat

NNT : 2023UPASG058

université PARIS-SACLAY
ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre: Traduction Automatique de Contenus Générés par l'Utilisateur : une Évaluation des Systèmes de Traduction Neuronaux dans des Conditions Zero-shot

Mots clés: traduction automatique, contenu généré par l'utilisateur, langue non canonique, traitement automatique de la langue.

Résumé: Les avancées rapides des télécommunications au cours des dernières décennies ont révolutionné la manière dont les gens échangent des informations. Grâce à ces progrès, l'utilisateur moyen peut désormais communiquer avec d'autres personnes à travers le monde en temps réel et avec un délai minimal. Avec environ 60% de la population mondiale ayant accès à Internet, des milliards d'individus interagissent en partageant du contenu généré par les utilisateurs (UGC) sous diverses formes. Ce type de contenu, qui comprend souvent des critiques et des opinions, constitue une source précieuse d'informations, offrant une vue d'ensemble des tendances mondiales. La traduction automatique joue un rôle vital en permettant une communication fluide et en facilitant le traitement automatique de l'UGC à des fins d'exploration de données.
Cependant, la traduction des UGC présente des défis uniques par rapport à la traduction d'un texte traditionnel. L'UGC est très productif et présente divers phénomènes tels que des caractères répétés, des erreurs typographiques, des contractions, du jargon et des structures de phrases non conventionnelles. Ces spécificités entraînent un nombre important de mots hors vocabulaire (OOV) et de séquences rares, qui posent des problèmes car ils ne sont pas représentés de manière adéquate dans les corpus parallèles standard utilisés pour entraîner les modèles de traduction automatique. En outre, les techniques conventionnelles d'adaptation au domaine, telles que le "fine-tuning", n'ont qu'un succès limité dans la résolution de ces problèmes. Elles souffrent d'une dégradation des performances lorsqu'elles sont appliquées aux données du domaine et ne sont pas en mesure de suivre l'évolution constante de la nature de l'UGC.
Dans cette étude, nous nous concentrons sur la tâche de traduction automatique des UGC dans le scénario "zero-shot", où nous nous abstenons d'utiliser des données d'apprentissage spécifiques aux UGC. Notre objectif est de développer des architectures de traduction automatique plus généralisées, capables de gérer le "distributional shift", inhérente à l'évaluation de la traduction des UGC. Dans la phase initiale de notre recherche, nous avons consacré nos efforts à l'identification et à la quantification des spécificités de l'UGC qui entravent la performance de la traduction. Nous avons également créé des cadres d'évaluation et des collections de données pour nous aider dans cette tâche. À l'aide de modèles "off-the-shelf", nous étudions les difficultés rencontrées par les systèmes de traduction automatique lorsqu'ils traduisent des UGC et nous établissons un lien entre les erreurs et les mécanismes sous-jacents.
Ensuite, nous nous penchons sur l'étude et la proposition de différentes méthodes pour relever les défis posés par l'UGC. Ces méthodes comprennent l'exploration des pipelines de normalisation, l'emploi de techniques de tokenisation plus granulaires et l'utilisation de modèles de variables latentes pour améliorer la robustesse des systèmes de traduction automatique. Pour chacune de ces approches, nous évaluons systématiquement les performances et la robustesse des systèmes, nous effectuons une analyse détaillée des erreurs et nous proposons des pistes prometteuses pour aborder la traduction automatique des UGC dans une évaluation "zero-shot".

université PARIS-SACLAY
ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Title: Machine Translation of User-Generated Contents: an Evaluation of Neural Translation Systems under Zero-shot Conditions

Keywords: machine translation, user-generated content, non-canonical language, natural language processing.

Abstract: The rapid advancements in telecommunications over the past few decades have revolutionized the way people exchange information. Thanks to these advancements, the average user can now communicate with others across the globe in real-time and with minimal delay. With approximately 60% of the global population having Internet access, billions of individuals interact by sharing user-generated content (UGC) in various forms. This kind of content, which often includes reviews and opinions, provides a valuable source of information, offering a comprehensive view of global trends. Machine Translation (MT) plays a vital role in enabling smooth communication and facilitating the automatic processing of UGC for data mining purposes.

However, translating UGC presents unique challenges compared to translating traditional text. UGC is highly productive and exhibits various phenomena such as repeated characters, typographical errors, contractions, jargon, and unconventional sentence structures. These specificities lead to a significant number of Out-of-Vocabulary tokens (OOVs) and rare sequences, which pose problems since they are not adequately represented in the standard parallel corpora used to train MT models. Additionally, conventional domain adaptation techniques like fine-tuning have limited success in addressing these challenges. They suffer from performance degradation when applied to in-domain data and are unable to keep up with the ever-evolving nature of UGC.

In this study, we focus on the task of automatically translating UGC in the zero-shot scenario, where we restrain from using any UGC-specific training data. Our aim is to develop more generalized MT architectures that can handle the distributional drift inherent in UGC. In the initial phase of our research, we dedicated our efforts to identifying and quantifying the specificities of UGC that hinder translation performance. We have also created evaluation frameworks and data collections to aid in this endeavor. Using off-the-shelf models, we investigate the challenges faced by MT systems when translating UGC and link the errors to their underlying mechanisms.

Subsequently, we delve into the study and proposal of different methods to address the challenges posed by UGC. These methods include exploring normalization pipelines, employing more granular tokenization techniques, and utilizing latent variable models to enhance the robustness of MT systems. For each of these approaches, we systematically evaluate the performance and robustness of the systems, conduct a detailed error analysis, and offer insights into promising avenues for tackling the automatic translation of UGC in the zero-shot setting.

Acknowledgements

First and foremost, I want to thank my supervisors, Guillaume Wisniewski and Djamé Seddah, for the countless hours that they dedicated to our discussions and their deep involvement in this work. I am very fortunate for the opportunity to benefit from their expertise throughout my Ph.D., for their exceptional guidance and their continuous support, always looking out for me. This dissertation would not have been possible without them.

I would also like to thank the members of my dissertation committee, Thierry Poibeau, Christophe Cerisara, Joseph Le Roux and Anne Vilnat, for their invaluable feedback. Their expertise and diverse perspectives provided rich insights into this work.

I am deeply grateful to the LISN and Inria labs for making my time as a Ph.D. candidate such an enjoyable and enriching experience. I am extremely fortunate to have crossed paths with amazing and talented colleagues and researchers. I will always cherish the time I spent with Aina and Leonardo and I thank them for their companionship, valuable perspectives and help during this journey. I would also like to thank Christian for his kindness, friendship and unconditional support. I was also very lucky to met inspiring and experienced fellows it the beginning of my Ph.D., in particular, Matthieu, Rachel, Lauriane, Gaël, Pooyan and Yoann; I am thankful for their advice and for the enlightening discussions we had, which greatly contributed to this work. My profound gratitude also goes to Arij and Wissam, for their help, creative ideas and rich insights.

I must acknowledge the financial support provided by the ANR's ParSiTi project, without which this research would not have been possible. Additionally, I am grateful for the computational resources from the IDRIS (LabIA and Jean-Zay platforms) and the Inria's clusters (RIOC and CLEPS), which were instrumental in this research work.

The support from my friends was a determinant force during my Ph.D. experience. A

special mention goes to Natalio for his technical advice on the aesthetics of the figures in this dissertation.

Finally, I want to thank my family for their strong support and loving care, which have been the primary motivations behind my academic journey. I owe this work to them.

Summary

In this work, we address the automatic translation of user-generated content (UGC), that is to say, the content published by users on online platforms, such as social media discussion channels or public user reviews. Translating UGC is both a necessary step to make this data amenable to automatic processing and a way to test the robustness of NLP pipelines to this kind of content: the way we now communicate through social networks has resulted in a fundamental change in the quantity and fluidity of exchanges, which makes UGC a particularly rich source of information: by automatically processing these massive, unstructured, and diverse texts, we can provide insights of large-scale trends in – virtually – real-time if (and only if) we are able to cope with the extreme, ever-evolving lexical and morphological variability that characterize this kind of content.

At a time when neural networks and, more specifically, transformers, seem to provide a solution to all NLP problems, translating UGC also constitutes an important contribution to the question of identifying the limits of this architecture, a problematic currently at the heart of many works in the NLP community. Indeed, due its high productivity, its multi-lingual, language-specific and topic-specific nature, UGC is the epitome of expecting endless possibilities. Deploying MT systems that are capable of handling UGC is highly challenging as this kind of content is not present in the corpora usually considered to train MT systems and the ever-evolving nature of UGC, as well as the costs of cleaning and annotating UGC data do not allow us to build the parallel corpora necessary to learn or even to fine-tune a neural translation model. This is why we put ourselves in a particularly difficult zero-shot scenario likely to highlight the limits of NMT.

We start this dissertation by studying **UGC specificities** that make it different from standard and widely-available parallel corpora. In this respect, we investigate distinct UGC linguistic features that we later link to translation quality. We analyze and compare the prediction errors

that occur when translating UGC using different **off-the-shelf MT baselines**.

As a first approach to the task at hand, we explore UGC normalization using automatically extracted phonetization to take into account phonemic information not contained in the corpus. Moreover, **phonemic similarities** often lead to typographical errors and compressed/abbreviated phonemic forms, which we study and aim to cope with. Subsequently, we explore the robustness properties of MT to UGC when using **different translation granularities** (e.g. character-based or subword-based tokenization) and end-to-end data-driven tokenization rules. To assess the robustness of such models, we built an **UGC evaluation framework** that can isolate each of UGC characteristic to assess their individual impact on automatic translation. Finally, in the last part of this research, we explore, compare and propose MT architectures with more **robust learning representation** by means of **variational inference**, and study how such models can enforce generalization over UGC by alleviating overfitting.

Résumé

Dans ce travail, nous abordons la traduction automatique du contenu généré par l'utilisateur (UGC), c'est-à-dire le contenu publié par les utilisateurs sur des plateformes en ligne, comme les canaux de discussion des médias sociaux ou commentaires publiques d'utilisateurs. La traduction de l'UGC est à la fois une étape nécessaire pour rendre ces données exploitables pour un traitement automatique et un moyen de tester la robustesse des pipelines NLP pour ce type de contenu. En outre, en s'attelant à cette tâche, la technologie s'attaque à la barrière linguistique, permettant une conversation plus fluide entre les locuteurs du monde entier, tout en étant capable d'utiliser la langue avec laquelle les utilisateurs sont plus à l'aise. En effet, les réseaux sociaux ont changé notre façon de communiquer, permettant une portée et une fluidité d'échange sans précédent. Cela fait que l'UGC constitue une source d'information particulièrement riche : en traitant automatiquement ces textes massifs, non structurés et diversifiés, nous pouvons fournir des tendances à grande échelle en – virtuellement – temps réel si (et seulement si) nous sommes capables de faire face à l'extrême variabilité lexicale et morphologique, en constante évolution, qui caractérise ces textes.

En raison de sa productivité élevée, souvent multilingue, spécifique à une langue ou un ou spécifique à un sujet, et dont les acteurs sont des millions de pairs décentralisés à travers le monde, la traduction des UGC est l'exemple même de l'attente de possibilités infinies. Ceci constitue un scénario particulièrement difficile pour déployer des systèmes de MT capables de traiter le UGC car ses propriétés ne sont pas présentes dans les corpus standard, largement utilisés pour entraîner les systèmes de traduction automatique. De plus, la grande diversité des UGC et la présence de formes émergentes de communication, ainsi que les coûts de nettoyage et d'annotation des données de l'UGC, imposent de fortes contraintes à la production de grands corpus parallèles de ces textes pour l'entraînement ou l'adaptation au domaine.

Nous rapportons et analysons les principaux problèmes des architectures de MT actuelles lors du traitement de l'UGC et nous orientons notre étude pour aborder cette tâche sous une évaluation détaillée, en rapportant systématiquement les résultats sur des tests UGC qui ont une nature et source diverses, ainsi qu'en contraignant nos ressources d'entraînement à des corpus standards et bien formatés. Dans cette ligne de pensée, nous nous concentrons principalement sur les propriétés de robustesse des modèles de MT et visons à proposer des systèmes de traduction plus intrinsèquement robustes et généraux.

Nous commençons cette thèse en étudiant les **spécificités des UGC** qui le rendent différents des corpus parallèles standard et largement disponibles. A cet égard, nous étudions les traits linguistiques distincts de l'UGC que nous lions ensuite à l'impact sur la performance lors de la traduction de ce type de texte. Nous analysons et comparons les erreurs de prédiction qui se produisent lors de la traduction d'UGC à l'aide de différents systèmes **baselines de MT standard**.

Comme première approche de la tâche à accomplir, nous explorons la normalisation de l'UGC à l'aide de la phonétisation extraite automatiquement afin de prendre en compte les informations phonétiques non contenues dans le corpus. De plus, les **similarités phonémiques** conduisent souvent à des erreurs typographiques et à des formes phonétiques compressées/abrégées, que nous étudions et cherchons à résoudre. Par la suite, nous explorons les propriétés de robustesse de la MT à l'UGC lors de l'utilisation de **différentes granularités de traduction** et des règles de tokenisation apprises de bout en bout; et nous élaborons sur les capacités de vocabulaire ouvert des systèmes MT au niveau des caractères. Pour évaluer la robustesse de ces modèles, nous avons conçu un **cadre d'évaluation de l'UGC** qui peut isoler chacun des phénomènes de l'UGC annotés afin d'évaluer leur impact individuel sur la traduction automatique. Enfin, dans la dernière partie de cette recherche, nous explorons, comparons et proposons des architectures de traduction automatique qui ont pour but d'apprendre **représentations robustes** en utilisant des méthodes d'**inférence variationnelle** qui visent la généralisation sur l'UGC en réduisant l'overfitting.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter, we introduce the scope, questions, and contributions of this dissertation, before presenting the main scientific context in which this work takes place. We will start by describing the interest of correctly processing automatically User-Generated Content (UGC), that is to say the content published by users of online platforms, such as social media discussion channels. We then establish the set of questions that will be addressed throughout this dissertation, the outline of its contents, and the contributions of this study.

1.1 Context

In recent decades, forms and channels of communication have undergone a steep change following the revolutionary onset of the Internet and its associated technologies. This has given regular users access to a vast and increasing number of platforms and media for content exchange, and a reach and organized information sharing that is unprecedented in history. The large amount of data on the Internet needs to be automatically processed to take full advantage of it.

In this work, we specifically focus on written forms of User-Generated Content (UGC) and study how Natural Language Processing (NLP) methods perform when treating such a kind of text. More precisely, we focus on Machine Translation (MT), a fundamental task to tackle in order to facilitate and ensure information sharing between speakers of different languages.

Concretely, working on UGC MT is putting efforts towards making communication and information exchange more fluid between individuals, as millions of users interact, employing

hundreds of different natural languages and dialects every day. In addition, the immense and diverse volume of UGC can be regarded as a rich and ubiquitous source of data. As such, automatically processing these texts is of great interest both from an economic and societal point of view, as we will see in Chapter 3. Moreover, online communication catalyzed the rapid and massive emergence of new forms of texts accompanied by increased volume in multilingual user interaction. In this context, the "Parsing the Impossible: Translating the Improbable" (**ParSiTi**) project (ANR-16-CE33-0021), which funded this doctoral work, seeks to adapt current language technologies to cope with the challenges raised by UGC.

Indeed, most of the NLP research mainly focuses on clean edited high-resource texts, overseeing the non-canonical, multilingual and contextual nature of such information streams and the **ParSiTi** project, identifies two main scientific challenges related to the automatic processing of UGC. The first one is **the lack of generalization of current NLP models**, which emerges from the shallow or void learning of linguistic knowledge in state-of-the-art NLP models. As these models consider tokens (i.e. phrases, words our sub-word units) as unstructured discrete units, they will consider the UGC French sentence "*ki ca ?*" and its canonical version "*qui ça ?*" as two unrelated sentences, whereas the average French-speaking user could easily identify the similarities (phonemic resemblance for the first token, "*ki*", and lack of cedilla for "*ca*"), and thus correctly interpret the phrase. There are also **multilingual and strongly contextual utterances in UGC**, which can, for instance, be seen in Table 1.1, where understanding the user's discussion is challenging without the context given by the photo.

Specifically, the main focus of this work, UGC automatic translation, is an important problem for open and real-world MT systems, which must be able to cope with such arbitrarily diverse text constructs that ultimately trigger the apparition of out-of-vocabulary (OOV) tokens such as non-existent words or foreign alphabets. The lack of MT robustness has led to unfortunate mistranslations (sometimes at large scale) in the past, as we will discuss further in Chapter 3.

1.2 Our approach

In this research work, we intend to design MT architectures that are robust to out-of-distribution (OOD) texts. More specifically, we will consider two kinds of OOD texts: out-of-domain canonical texts and noisy UGC. As we will discuss in Chapter 3, UGC is a multi-domain type of text with

(@rigolboche)

Original source	Bing© translation
→ T'as vu il l'a bien cherché wsh #AperoChezRicard	→ Did you see he looked for it wsh #AperoChezRicard
→ +10000, shah!	→ +10000, shah!
→ tabuz, lavé rien fé	→ tabuz, washed nothing fe
→ ki ca ? lemecousonchien ?	→ ki ca ? theguyorhisdog?
→ Wtf is wrong with him ? #PETA4EVER	→ Wtf is wrong with him ? #PETA4EVER.
→ ki ca ? le chien ? looool	→ ki ca ? the dog? looool

Figure 1.1: Typical social media thread initiated by a seed photo and its automatic translation. *Inspired from a real conversation about a series of demonstrations that took place on Greece. Bing was used (on the 14/05/2015) as it was then the official MT engine for Twitter and Facebook.*

many specificities.

To achieve this goal, we will use, throughout this work, a consistent experimental protocol in which we systematically evaluate our models on both canonical OOD (i.e. different domain from the train data), and OOD UGC test sets. Such an evaluation protocol, in addition to reporting results of all our systems on in-domain test sets, is intended to assess whether the models developed for translating UGC are still able to translate canonical corpora.

Furthermore, since we aim to improve robustness to a wide set of OOD conditions, we decided not to use any target-specific information, and we have restricted ourselves to develop only zero-shot methods. Specifically, we assume our systems to have no access to UGC during training. Indeed, because of the vast diversity of UGC specificities, it is impossible to collect all possible variations in a single corpus and UGC always contains new emergent forms of written expression: for instance, Martínez Alonso et al. (2016) shows that normalization schemes designed for UGC data collected in 2011 were not suitable for data collected in 2014. It is therefore necessary to develop models that perform well, not only on a given UGC corpus, but that will continue to do so in the future or under different circumstances (platform, language,

domain, etc.) without constantly annotating data, which is a particularly challenging task. That is why, our study focuses on characterizing and addressing UGC performance from a robustness perspective rather than using target-distribution information, for instance, by fine-tuning pre-trained models on UGC data. Modeling and explaining MT performance on noisy UGC remains today a challenge. To address this aspect, we introduce an extensive UGC evaluation framework to assess the detrimental impact of UGC's distinct specificities. It is noteworthy that, when this doctoral research work started, a lack of UGC evaluation resources in the literature prevails, making it difficult to study their impact on NLP tasks. Since, a rather limited number of parallel corpora and evaluation frameworks (including ours) have been proposed. The parallel test set annotated with UGC specificities we developed during this thesis in Chapter 6 is one of the main contributions of this work.

1.3 Challenges raised by UGC: questions to address

Now that the scope of our work has been introduced and that we have stated the premises and interests of this study, we will detail the plan of this report and our main contributions. In a first step, we identify and quantify the impact of UGC specificities on state-of-the-art MT systems. Then, we propose and discuss different approaches to address these problems, ranging from pre-processing and normalization via a phonetization process, to considering different NMT architectures that could deal with the noise present in our evaluation. In all of our experiments, we systematically show an error analysis to highlight and explain the caveats and the observed improvements of our methods.

Now we describe the research questions raised by the translation of UGC that we address in this work.

How does UGC impact translation quality? At first, we want to evaluate the impact of UGC specificities on MT translation quality. To do so, we review, in Chapter 2, popular state-of-the-art MT architectures. We have considered both classical phrase-based models and models based on neural networks. Our first results raise several questions:

- What is the difference between phrase-based and neural machine translation systems, and how comparably robust are they to UGC?

- How do NMT systems behave when processing such kinds of texts and can translation errors be linked to their architecture?

In Chapter 4, we compare the MT predictions of UGC and quantitatively assess the performance of different MT systems and evaluate their robustness. In the second part of this chapter, we study the internal working of NMT systems to explain some of the observed errors as well as models behavior.

Which UGC specificities result in translation errors? The work described in Chapters 4 and 6 aims to characterize which of the specificities, described in Chapter 3, make translation more complicated, thus leading to a lower translation quality. More precisely, we address the following research questions:

- Do UGC's specificities distribution conditions performance for a given NMT architecture?
- How much do such UGC specificities impact translation quality individually?

In this work, the potential interactions between different types of UGC are addressed carefully by isolating and documenting translation quality detriment caused by them.

What kind of methods can be used to cope with UGC specificities for zero-shot translation? To answer this question, we propose and explore normalization methods and promising NMT architectures to overcome the variability of UGC due to its inherent links to users' creativity. In the scope of this work, we keep a systematic effort toward studying zero-shot methods, i.e. without using neither parallel nor monolingual UGC. Some questions addressed in this part of the thesis are:

- Can supplementary natural language information, such as phonetization, provide a solution to the challenges raised by the translation of UGC?
- Can subword-level NMT models' properties take advantage of morphological information to resolve OOV occurrences imposed by UGC?
- Does the use of latent variables in translation models reduce the impact of the UGC noise on the translation quality?

Under the zero-shot translation condition, we intend to review methods that augment the capacity of MT models by either providing information, as the model described in Chapter 5 that makes

use of automatically-extracted phonetization or leveraging on subword morphological features, as we explore in Chapter 6. Finally, in Chapter 7, we explored latent-vector models, which, notably, take into consideration stochastic perturbations to the model, and we give insights on their robustness to UGC.

1.4 Contributions of this work

Addressing these questions and analyzing the experimental evidence that they resulted in, resulted in several contributions:

- The caveats and considerations for the range of different MT systems when processing UGC were noticed and documented, including the effects that this noise has on NMT, specifically, the attention mechanisms. Additionally, we investigate and report a robustness perspective of such NMT models that we further link to specific morphosyntactic specificities in this kind of texts.
- A new automatic normalization pipeline was developed to account for phonetic writing and letter omission by leveraging pronunciation similarities in French, which proved to be especially effective for aiding in the translation of UGC short sentences.
- The effect of translation level for translating noisy texts was studied, and we describe new considerations regarding the open-vocabulary capabilities of character-based MT involving the choice of vocabulary and show that it is possible to reduce the effect of rare or OOV characters in MT.
- A series of corpora and evaluation protocols were designed specifically to unveil the impact of the different UGC specificities introduced in Chapter 3 on translation quality. In order to do so, we released an evaluation framework to control the UGC specificity present during evaluation, notably preserving only one specificity at the time to disentangle their individual effects on performance. This resource also enabled us to account for the performance impact of a significantly wider range of such specificities than that was reported in the state-of-the-art literature, and provided a new and extensible framework for evaluating the robustness capabilities of any MT system.
- We explore models that use latent representations by including variational Bayesian methods in state-of-the-art MT models (namely Transformer) in order to assess whether

such approaches can produce inherently more robust translation systems from noisy inputs. We also propose a novel variational MT architecture that achieves better results compared to a strong baseline model, especially, when processing UGC. We show how these methods act as noise regularizer, enforcing the model to learn more robust representations. We also study and showcase the semi-supervised training capabilities of these models by introducing a source-side sentence reconstruction loss term. Finally, we conduct translation experiments with our proposed model in a different language pair (Japanese-English) and show that our findings also hold in this experimental setup.

1.5 Dissertation outline

In this work, we study the impact of UGC on MT and characterize this type of text in order to link translation quality to specific morphosyntactic phenomena presented in UGC. Then, we propose and discuss methods to cope with the noise present in UGC to bridge the performance gap between such texts and canonical texts.

We start by describing the Phrase-Based Statistical Machine Translation (PB-SMT) and NMT models we consider in this work. Furthermore, we discuss and explain MT notions and methods traditionally used to cope with noise and domain-drift scenarios by leveraging segmentation and translation granularity. We also explain the automatic metrics used to assess the performance and robustness capabilities of the MT systems used in this work.

We then discuss, in **Chapter 3**, the interest and challenges raised by UGC for NLP models, specifically translation models. We also explain, in this chapter, the importance from a user perspective of correctly processing UGC to extract useful information about a wide range of topics. We additionally highlight the importance of devising robust MT models for real-world applications, as they can hardly avoid UGC when deployed in public and real-world scenarios and applications.

Once the context surrounding this doctoral thesis, the difficulties to overcome when translating UGC, and the main MT and evaluation methods are introduced, we develop an in-depth study of our first MT baselines in **Chapter 4**. In this part, we describe the caveats of different MT systems when they are processing UGC, including the effects that this type of texts has on NMT architectures, specifically, the attention mechanisms. Additionally, we investigate and report a

robustness perspective of the studied MT systems and discuss how different MT paradigms are sensitive to noise in the input.

In order to set a first baseline for improving UGC MT, we explain, in **Chapter 5**, our approach to address the phonetic writing specificities of French UGC such that we can automatically generate token-level normalization to reduce the presence of noise in the system input. We report improvement over several UGC test sets while hurting performance on canonical tests because of the artificial ambiguities induced by the method. We also report how sentence length impact the performance of our methods.

We continue our search for improving modeling by studying the impact of translation granularity on MT robustness in **Chapter 6**. The motivation of the method developed in this chapter is to improve the capacity of NMT to learn token representation to representations to ensure that noisy tokens and their normalized form have "close" or "similar" representation, rather than pinpointing a UGC specificity such as we did in Chapter 5. By doing so, we aim to attack a broader range of UGC phenomena, thus leading our research to rigorously assess performance and robustness in order to know what our models are doing better.

In the same track of thought, although the robustness of different types of MT systems was assessed, the reasons why UGC translation performs typically worse than canonical evaluation sets, remains, at best, obscure and corpus-level metrics have proven to be too coarse of a metric to explain the impact on the performance of the variety of UGC features described in Chapter 3. This is why, in Chapter 6, we have designed an evaluation framework that, to the best of our knowledge, contains the widest range of UGC specificity typology in the literature with 13 different UGC specificity annotations. Also, to address the potential interaction between such UGC phenomena, we designed a code base that can isolate them and generate a series of subcorpus with any given distribution of such types. Such an evaluation framework proved useful to compare different MT systems, concretely, NMT models with finer translation granularities (subword and character segmentation), which could benefit from a larger token-level coverage with a reduced number of vocabulary elements. Indeed, by using our proposed evaluation resource, we were able to assess robustness properties of the character-based NMT models in comparison to our baseline vanilla NMT systems, where the former showed to be considerably more robust to misspellings (character changes, missing diacritics), tokenization errors, graphemic and punctuation stretching, and inconsistent case changes.

Moreover, as character-level NMT showed the worst robustness capabilities when confronted to special and rare characters, we performed a study to revisit the open-vocabulary properties of such char-level MT systems and these results showed that we can get improved translation performance by correctly managing unknown character by the means of setting the character vocabulary size correctly, often overlooked in today's character-level MT literature.

As a final stage of this work, in **Chapter 7**, we investigate another approach to zero-shot noisy UGC, for which we have introduced and studied new Variational Neural Machine Translation (VNMT) architectures that outperformed strong baselines when translating UGC. In this phase chapter of the dissertation, we provide further insight on how VNMT latent neural representations are more robust to UGC, by proposing novel approaches of visualization and explanation surrounding such models. In addition, we conducted a series of detailed ablation experiments to justify and understand the impact of our design choices.

Finally, in **Chapter 8**, we conclude and give some perspectives and interesting future avenues to cope with UGC MT that emerged from this work.

1.6 Publications related to this thesis

- José Rosales Núñez, Djamé Seddah, Guillaume Wisniewski (2019). Comparison between NMT and PBSMT Performance for Translating Noisy User-Generated Content. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics* (NoDaLiDa 2019), October 1-2, Turku, Finland. (Rosales Núñez et al. 2019) **(Chapter 4)**.
- José Rosales Núñez, Djamé Seddah, Guillaume Wisniewski (2019). phonetic Normalization for Machine Translation of User Generated Content. In *Proceedings of the 5th Workshop on Noisy User-generated Text* (W-NUT 2019), November 4th, Hong Kong. (Núñez et al. 2019) **(Chapter 5)**.
- José Rosales Núñez, Guillaume Wisniewski, Djamé Seddah (2021). Noisy UGC Translation at the Character Level: Revisiting Open-Vocabulary Capabilities and Robustness of Char-Based Models. In *Proceedings of the Seventh Workshop on Noisy User-generated Text* (W-NUT 2021), November 11th, Punta Cana, Dominican Republic. (Rosales Núñez et al. 2021b) **(Chapter 6)**.
- José Rosales Núñez, Djamé Seddah, Guillaume Wisniewski (2021). Understanding

the Impact of UGC Specificities on Translation Quality. In *Proceedings of the Seventh Workshop on Noisy User-generated Text* (W-NUT 2021), November 11th, Punta Cana, Dominican Republic. (Rosales Núñez et al. 2021a) **(Chapter 6)**.

- José Rosales Núñez, Djamé Seddah, Guillaume Wisniewski (2023). Multi-way Variational NMT for UGC: Improving Robustness in Zero-shot Scenarios via Mixture Density Networks. To appear in *Proceedings of the 22nd Nordic Conference on Computational Linguistics* (NoDaLiDa 2023), May 22nd-24th, Tórshavn, Faroe Islands. **(Chapter 7)**.

Chapter 2

Machine translation: a review of methods and techniques

Now that we have introduced the problems and questions we will be addressing in this dissertation, we present the different Machine Translation frameworks and methods used in our work. We will start by describing the two most popular families of MT approaches: phrase-based statistical systems (Section 2.1) and neural architectures (Section 2.2), which are the systems used in the experiments throughout this work. Then, in Section 2.3, we will describe and discuss MT tokenization methods as well as other approaches and considerations introduced to address the Out-of-Vocabulary (OOV) problem, notably present in out-of-distribution evaluation. In the final part of this chapter (Section 2.4) we discuss different methods to evaluate the performance of the models and explain their behavior.

2.1 Statistical MT

Statistical Machine Translation (SMT) (Weaver 1949; Brown et al. 1988) is a set of methods that rely on statistics to translate sentences by modeling the relationship between two parallel bilingual corpora without the need to use explicit linguistic rules.

Noisy-channel paradigm In SMT, the problem of choosing the best translation $\hat{y}$ given the input sentence $\boldsymbol{x}$ can be formally formulated as the following optimization problem:

$$\hat{y} = \arg\max_{\boldsymbol{y}} p(\boldsymbol{y}|\boldsymbol{x}) = \arg\max_{\boldsymbol{y}} p(\boldsymbol{x}|\boldsymbol{y}) \times p(\boldsymbol{y}) \tag{2.1}$$

in which $p(\boldsymbol{x}|\boldsymbol{y})$ is estimated by the *translation model*, which will be introduced next, and $p(\boldsymbol{y})$, computed by the target *language model*, a well-known probabilistic model that captures dependencies between words of a given sentence. This framework combining translation and target language models is known as the *noisy-channel model* (Shannon 1948).

It should be noted that the probability search space of translations is considerably large as, even if we have the n correct target words, there will be $n!$ permutations, and finding the best-scoring permutation (i.e. solving the $\arg\max$) is NP-complete (Udupa and Maji 2006). That is why SMT relies on approximate search methods, such as beam search, to find the best translation.

Phrase-based approach Phrase-based statistical machine translation (PB-SMT) is the approach in this framework that achieves the best performance and had long been the state-of-the-art method in SMT (Koehn 2009). Phrase-based models process *phrases* made of sequence of words (i.e. n-grams) that are translated using a translation table, containing, for each sequence of source words of the training data, the possible corresponding target word sequences and their probabilities. For instance, the translation table associates the German word "*natürlich*" to its possible translations: "*of course*", "*of course ,*" and "*naturally*". The phrase table is built automatically from the word alignments.

In a phrase-based model, the source sentence is broken down in K phrases x_k and the probability $p(\boldsymbol{x}|\boldsymbol{y})$ is estimated as:

$$p(\boldsymbol{x}|\boldsymbol{y}) = \prod_{k=1}^{K} \phi(x_k|y_k) \tag{2.2}$$

This equation uses the probability $\phi(x_k|y_k)$ of translating y_k to x_k that is stored in the translation table, automatically extracted from the training data (Koehn 2009). This first (simplified) model has been devised to allow taking into consideration other features (e.g. to describe words

reordering) in the estimation of the translation probability (Bertoldi et al. 2009). Equation 2.2 highlights the main characteristics of phrase-based models: computing the score of a translation hypothesis relies on finding an exact match of a source phrase (i.e. a subsequence of the source sentence) in the phrase table, which is particularly problematic for the translation of UGCs since the high variability present in the source sentences strongly reduces the probability that a given sequence of words has been seen during training and ultimately extracted in the phrase table.

2.2 Neural machine translation

Neural Machine Translation (NMT) is a set of methods that employs artificial neural network architectures to translate sentences. These models are capable of automatically learning relevant representations of the source and target sentences from the training set and do not rely any longer on exact matches of source phrases as PB-SMT systems. The MT task is part of a wide family of architectures called *sequence-to-sequence* models that can be used to learn the mapping of an observations' sequence to a label sequence, and it is behind multiple other NLP tasks, such as language modeling and summarization. NMT has proven to outperform statistical MT models, setting a new state-of-the-art for MT when compared to SMT approaches (Bentivogli et al. 2016; Mutal et al. 2019).

In this section, we review the most popular NMT architectures that we used in our experiments. First, we describe the encoder-decoder model, which is at the heart of NMT and other sequence modeling tasks. We then focus on two neural frameworks, namely, RNN-based models and attention-based models (i.e. *Transformers*), which are the two most popular NMT architectures nowadays. Both architectures generate a hidden representation, either sequentially in recurrent networks or in parallel in Transformers. This representation is used to encode the meaning of the source sentence, and passed in cascade through the network to generate the target sequence of target language tokens.

2.2.1 The encoder-decoder architecture for MT

Most of the modern NMT models have adopted the encoder-decoder paradigm, introduced by Sutskever et al. (2014) and Cho et al. (2014). Using this paradigm in MT has two main advantages over previous approaches: first, it generates contextually appropriate translations

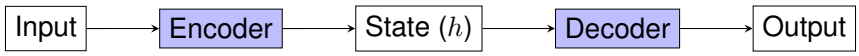


Figure 2.1: The encoder-decoder architecture..

(https://d2l.ai/chapter_recurrent-modern/encoder-decoder.html)

(Way 2019) for target sentences with an arbitrary length (Jurafsky and Martin 2009); second, it allows designing and train a single, end-to-end model directly from a parallel corpus.

The underlying idea of this paradigm is to have a first — possibly stack of — neural networks, known as the *encoder*, which outputs, for each token of the T_S-length source sentence $\boldsymbol{x} = (x_1, x_2, ..., x_{T_S})$, a contextualized representation $\boldsymbol{h} = \left(h_1^D, h_2^D, ..., h_t^D, ..., h_{T_S}^D\right)$ each of these representations being a vector of D dimensions, where D is generally called the "hidden size". These vectors (or only some of them) are then 'fed' to another neural network (i.e. considered as its input), called the *decoder*, that is in charge of generating the T_d-length sequence of target tokens $\boldsymbol{y} = (y_1, y_2, ..., y_n)$. An overview of this type of architectures is shown in Figure 2.1.

2.2.2 Sequential RNN-based models

The first type of NMT models that were proposed are based on Recurrent Neural Networks (RNN). This kind of architecture processes the input sentence token by token and progressively builds a "hidden vector", $\boldsymbol{h}$, as introduced previously. For a given encoding step (h_t^d, where $t \in [1...T_S]$ represents each one of the source sentence tokens), depends on the previous tokens $1, ..., t-1$, allowing the RNN to propagate the information throughout the sentence and these representations to be contextualized. The last hidden vector contains the information that summarizes the whole source sentence and is used as the input of the decoder. In addition, bidirectional RNN encoders (Schuster and Paliwal 1997) have been shown to consistently outperform uni-directional encoders (Graves et al. 2005; Sundermeyer et al. 2014) by building two $\boldsymbol{h}$ vectors using the left-to-right and right-to-left streams (i.e. processing the source sentence from first to last token, as well as from end to start). It is worth mentioning that the decoder is always uni-directional since we do not have access to future target tokens, as they are generated one by one conditioning each decoding step to the past produced tokens.

In Figure 2.2, we show a typical encoder-decoder RNN MT architecture featuring a bidirectional encoder. Left-to-right and right-to-left encoding streams are marked as *forward* and *backward*, respectively. In a first stage, each token of the input sequence, $\boldsymbol{x}$, is passed to each

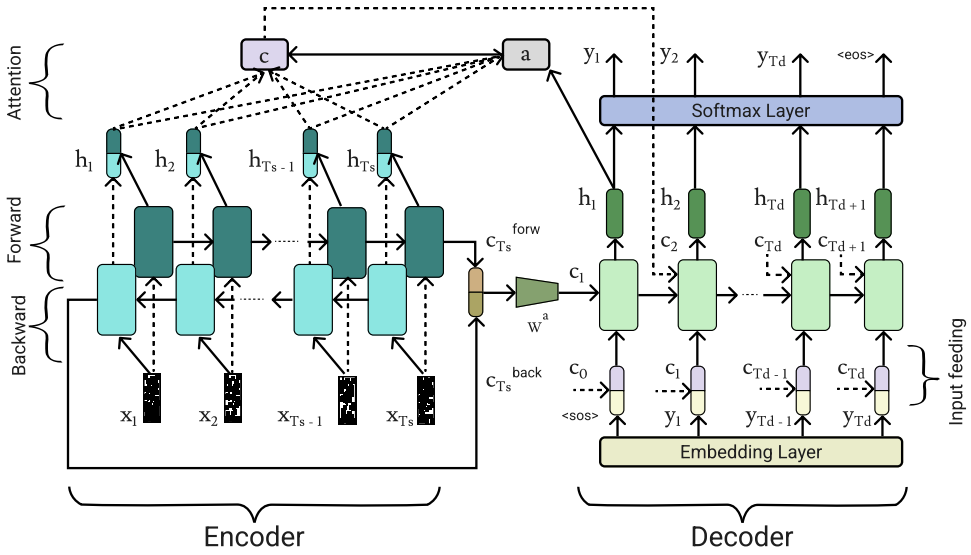


Figure 2.2: Bidirectional RNN Encoder-Decoder architecture. Reproduced from Chowdhury and Vig (2018).

of the encoder's RNN cells, whose hidden states ($\boldsymbol{h}_t$ for t in $[1, T_S]$, corresponding to the T_S-long source), are sequentially computed and propagated through these cells until the last state, $\boldsymbol{h}_{T_S}$, is generated. In addition, as discussed previously, the token-by-token encoding is usually done in a bidirectional fashion, and the final hidden vector is defined as the concatenation of the resulting representations for both directions ($[\boldsymbol{c}_{T_S}^{forw}; \boldsymbol{c}_{T_S}^{back}]$ in the figure). Figure 2.2 portrays this approach, showing left-to-right flow (forward) in dark green, and right-to-left (backward) in light green.

In a second stage, the decoder generates each of the target tokens in an autoregressive fashion, i.e. by outputting one token at a time, conditioned on both the context vector and the last token produced by the decoder, while propagating the inner hidden state of each of the decoder's RNN units. Finally, a *softmax* layer maps the RNN outputs to a probability distribution over each possible token of the target language vocabulary for each decoding step.

The optimization problem for NMT, as well as it sequential autoregressive decoding can be expressed as:

$$\arg\max_{\phi} p(\boldsymbol{y}|\boldsymbol{x}) = \arg\max_{\phi} \prod_{i=1}^{J} p(y_i|y_{j<1}, \boldsymbol{x}) \tag{2.3}$$

where ϕ is the architecture's trainable parameters and J is a maximal output length, which is used to stop the generation in case the `<eos>` symbol is not generated for the $(\boldsymbol{x}, \boldsymbol{y})$ pair.

In Figure 2.2, red and green blocks represent the encoder and decoder RNN neurons, respectively. These are most commonly chosen (Yang et al. 2020) to be Long-short Term Memory (LSTM) networks (Hochreiter and Schmidhuber 1997), which have been shown to outperform vanilla RNNs (Chung et al. 2014; Lipton 2015). In Figure 2.3 we can see the three gated mechanisms introduced within an LSTM cell, which, in contrast to vanilla RNNs, do not store all the information in its state for a training sequence, but selectively learns to remember or forget (Skrlj et al. 2019).

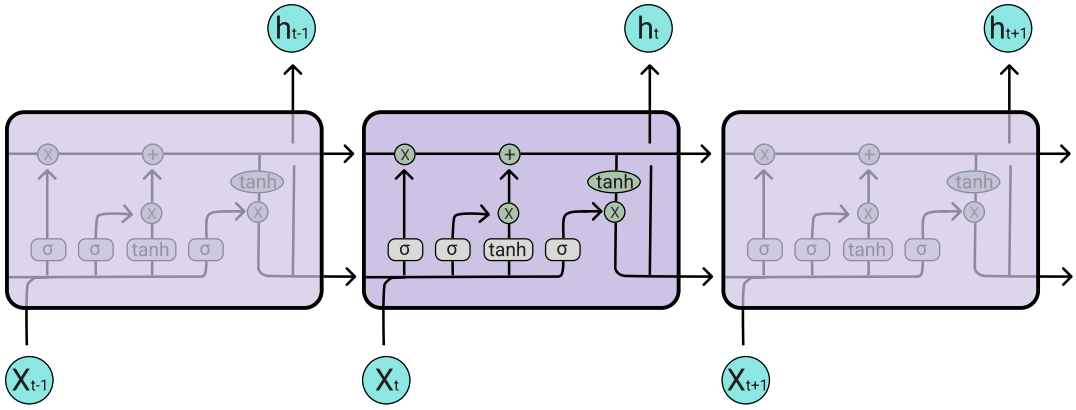


Figure 2.3: Diagram of an LSTM, the RNN architecture used in this work. Reproduced from (Olah 2015).

For all RNN-based NMT experiments presented in this work, we use the LSTM architecture, according to the current trends in the field (Yang et al. 2020). Nonetheless, to cope with the computational overhead due to longer sequences for the character-level RNN models in Chapter 6, we use a simplification of LSTM, consisting of a single gated output. It is known as Gated Recurrent Unit (GRU) (Cho et al. 2014), and it achieves similar performances to LSTM while being more computationally efficient (Chung et al. 2014).

Attention mechanisms As discussed in Section 2.2.1, the decoder network, is conditioned on the encoder's hidden vector $\boldsymbol{h}_t$, for each target token produced in an instant t according to the following equations:

$$\begin{aligned} h_t &= sigm(W^{hx}x_t + W^{hh}h_{t-1}) \\ y_t &= W^{yh}h_t \end{aligned} \tag{2.4}$$

where W^{hx} and W^{hh} are the encoder's parameters and W^{yh} the decoder's parameters.

Since the last hidden vector ($\boldsymbol{h}_{T_S}$) contains the representation for the whole input sequence, learning the alignments, i.e. the most important tokens in the input when producing a given target token; becomes a sub-task delegated primarily to the decoder network. In order to accomplish this, a method introduced by Bahdanau et al. (2015), called *attention*, aims to unburden the decoder by explicitly attributing the alignment modeling to the encoder. Since then, attention methods have greatly driven the research in NMT and proved to improve sequence-to-sequence architectures.

Concretely, at each decoding step $\mathtt{t}$, a context vector $\boldsymbol{c}_t$ is computed as a weighted vector along the $\boldsymbol{h}_{1:T_S}$ components of the hidden vector output by the encoder. This context vector is intended to help the decoder focusing more on a hidden vector component (that is, on a corresponding source token) than others, computed at each decoding step by the alignment model as:

$$c_i = \sum_{j=1}^{J} a_{ij} h_i \tag{2.5}$$

where $\boldsymbol{a}_t$ is the attention vector that assigns a certain weight to each token component of the hidden vector. The vectors $\boldsymbol{a}$ and $\boldsymbol{c}$ are shown accordingly in Figure 2.2. In order to calculate $\boldsymbol{a}_t$, a scoring function is combined with a *softmax* function to output a distribution on the last decoder's state $\boldsymbol{s}_{t-1}$ and the encoder's hidden vectors. This equation is formulated as follows:

$$a_t = softmax(scoring(\boldsymbol{s}_{t-1}, \boldsymbol{h})) \tag{2.6}$$

In this work, we study the attention behavior to perform error analysis and investigate whether such errors can be explained in terms of faulty attention weight distribution for a given decoding time step.

The concept of attention rapidly evolved and eventually led to a series of models in which the sequences and order of tokens can be fully captured by positional embeddings (Vaswani et al. 2017). This evolution resulted in a breakthrough in NMT research and in a new kind of architecture family: transformers.

2.2.3 Transformers

The Transformer model (Vaswani et al. 2017) demonstrates how self-attention networks can perform sequence-to-sequence tasks without explicitly modeling the sequential nature of a sentence, i.e. sequential decoding and training of the network's parameters in equation 2.4. Instead, the whole input token sequence is processed simultaneously in parallel and the position of each token is encoded by a positional embedding, where the several parallel and independent self-attention mechanisms provide information for each token of the input. Thus, Transformer self-attention layers connect the sequence tokens with a constant number of operations, whereas RNN networks require $\mathcal{O}(n)$ operations to for *n*-length sequences, resulting in improved performance of the former for long-range dependencies (Vaswani et al. 2017). As a consequence, the Transformer has access to more information simultaneously during the output generation than RNN approaches. Hence, this model has been widely adopted as the state-of-the-art architecture (Xia et al. 2019), as it has also been consistently shown to perform better than RNN. In addition, this model is highly scalable through parallelization, contrary to sequential architectures.

Transformer is another instance of the encoder-decoder paradigm, as shown in Figure 2.4. The principal component of this architecture is a set of multi-head self-attention networks, where several independent attention mechanisms model the inner relations between tokens

After reviewing the main MT architectures used in this work, we discuss other complementary methods that influence translation quality. Specifically, we will describe tokenization methods and granularity of translation, i.e. using words, subword or characters in the input and output vectors, which have a noticeable impact on translation quality (Sennrich et al. 2016) and play an important role in the vocabulary coverage learned by the MT model (Ataman et al. 2019).

2.3 Tokenization methods

In this section, we introduce the tokenization methods used during this work and discuss the motivation for using different token granularities in MT. Tokenization is especially important when there are many out-of-vocabulary (OOV) tokens, that is to say, tokens that have not been seen during training. This happens when translating morphologically rich or low-resource language or

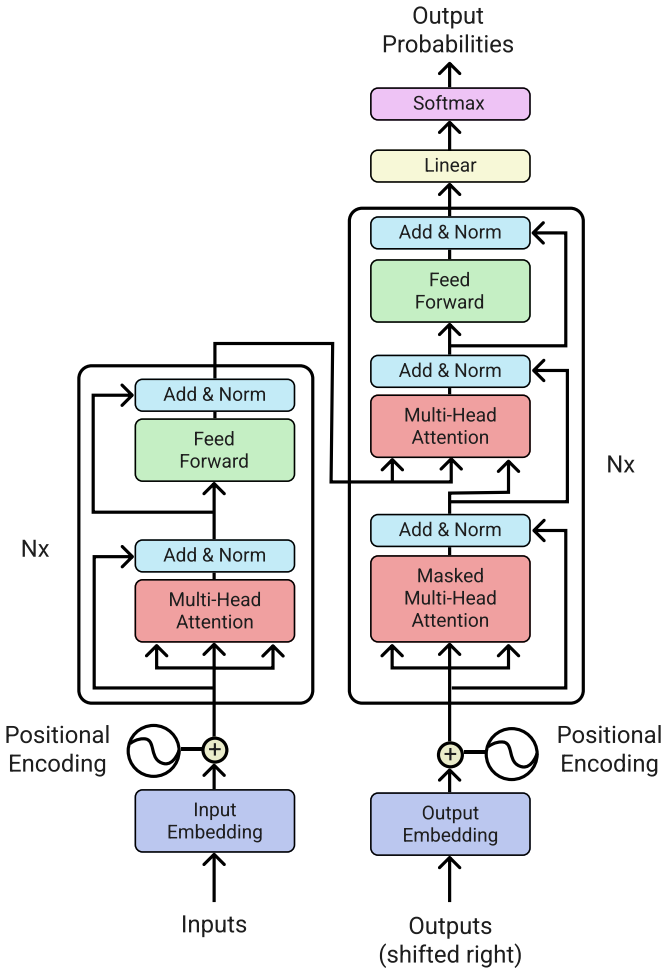


Figure 2.4: The Transformer Encoder-Decoder architecture. The encoder on the left side of the diagram and the decoder on the right one. Reproduced from Vaswani et al. (2017).

in the context of distributional shifts such as UGC translation. Coping with these OOV tokens is an important problem, as they often degrade translation performance (Sutskever et al. 2014) and tokenization is the simplest way to reduce their number.

MT systems were initially designed using words as the basic input unit: A rule-based tokenization algorithm appends spaces between independent sequence of characters (such as text and punctuation) in order to reduce vocabulary and make training easier (Ataman and Federico 2018). However, this approach results in many OOV because of the considerable lexical diversity of words in many languages and the poor compositional properties of many models. Translating these OOV raises many challenges as these words cannot be properly modelled, affecting translation quality. Indeed, OOVs have to be mapped to a special `<UNK>`

symbol that is generally kept unchanged during translation and is therefore still present in the MT output. This token can be directly copied into the output by incorporating a copy mechanism (Gu et al. 2016), or subsequently translated by replacing it in a post-processing step (Jean et al. 2015; Mi et al. 2016), or via embedding substitution (Haddad et al. 2018; Yang et al. 2018).

The "post-processing" approach is straightforward and consequently, by far, the most widely used in today MT systems (even if it often seems to be ignored in the literature with the development of NMT). Its main advantage is that this approach is completely independent of the MT architecture used and, thus, can be easily applied to any MT system. The alignment between the source sentence and the translation hypothesis can either be predicted directly using a word-alignment method such as IBM models (Mi et al. 2016) or can be deduced from the attention matrix as proposed by Jean et al. (2015). Throughout this work, we will use the former approach and replace `<UNK>` in the evaluation translation hypothesis using source-target alignments predicted with *fastalign* (Dyer et al. 2013).

This strategy is, however, very limited and several alternatives have been proposed to actually translate the OOVs rather than copying them unchanged from the source sentence. The most straightforward way to deal with OOV is to consider smaller, sub-word tokens and to leverage on composition to represent words (Sennrich and Zhang 2019).

The finest grain segmentation possible consists in cutting out the input sentence in a stream of characters and modeling it directly (resulting in a character-based model). This approach raises two main problems, the first being the computational cost of the resulting methods: the number of characters in a sentence is substantially larger than the number of words, and thus the computation times are mechanically increased. For instance, Luong and Manning (2016) report spending about 3 months training a purely character-level model. Another important problem, and a fundamental drawback of character-based models, is the difficulty of modeling long-range dependencies (Wood-Doughty et al. 2018; Al-Rfou et al. 2019).

To avoid the pitfalls of character models, subword-level segmentation models have been proposed: Their goal is to uncover a word segmentation that limits the number of units created (to avoid complexity problems) while limiting the size of the vocabulary (to ensure that the frequency of units is sufficiently high). Notably, Sennrich et al. (2016) propose to use the Byte-Pair Encoding (BPE) algorithm to produce a subword vocabulary and Schuster and Nakajima (2012) introduce `Wordpiece`.

The use of subword-level segmentation has proved to consistently outperform word-based MT and is currently systematically used in all NMT systems. We adopt BPE tokenization throughout this work unless stated otherwise (our character-based models in Chapter 6). When using BPE, the vocabulary is built by merging the most frequent sequences of characters to reach a number of operations set as a hyperparameter. The intuition is that frequent n-gram structures constitute subword particles (e.g. prefixes or suffixes) and, by leveraging on their compositionality, we can decompose potential OOV during evaluation in a sequence of in-vocabulary n-grams.

BPE ensures that there are almost no OOV tokens in both the train and test sets, with the exception, for instance, of words in the test set that contain characters that do not appear in the train set. However, translation quality still suffers from the actual OOV words that existed before BPE segmentation (Araabi et al. 2022), as these words are often divided into many BPE tokens.

2.4 Analyzing translation hypothesis

In this work, we will use two methods to analyze the predictions of the different MT systems we consider: the first method is a qualitative method that uses the attention matrices at the heart of NMT models to explain the decisions made by the translation system; the second method relies on the automatic metrics usually used to evaluate the translation systems. Finally, we will also introduce several metrics to assess translation robustness to noise in the input sentence that we use throughout this work.

2.4.1 Visualization: understanding NMT decisions

Although NMT systems are commonly regarded as black-box systems because their decision process is difficult to interpret, a popular way to gain insight into the inner workings of these systems is through self- and cross-attention matrices. Using attention to explain neural network decisions is subject to a long-lived debate (Serrano and Smith 2019), but has been proven to be informative in many works (Ding et al. 2017). The attention matrix quantifies the “importance” of a (source or target) token to generate each one of the target tokens. We will use this method in Chapter 4 to highlight some of the pitfalls of NMT systems when translating UGC.

Source-Target Attention Figure 2.5 gives an example of a cross-attention matrix (as the one shown in Figure 2.2) between an English source sentence (horizontal axis) being translated into French (vertical axis shows the predicted target tokens). It shows how the decoder focuses on some specific source words to different degrees when producing a given target word. Attention weights from Equation 2.6 are depicted in grey-scale (the whiter, the closer to 1). These are computed by the model as the attention scoring function is learned during training. For instance, it can be noted that to generate the French tokens “ne peut plus” the decoder mainly relies on the English tokens “can longer produce”. This shows how attention can give special importance to the most pertinent encoder hidden vectors (each corresponding to a source-side token).

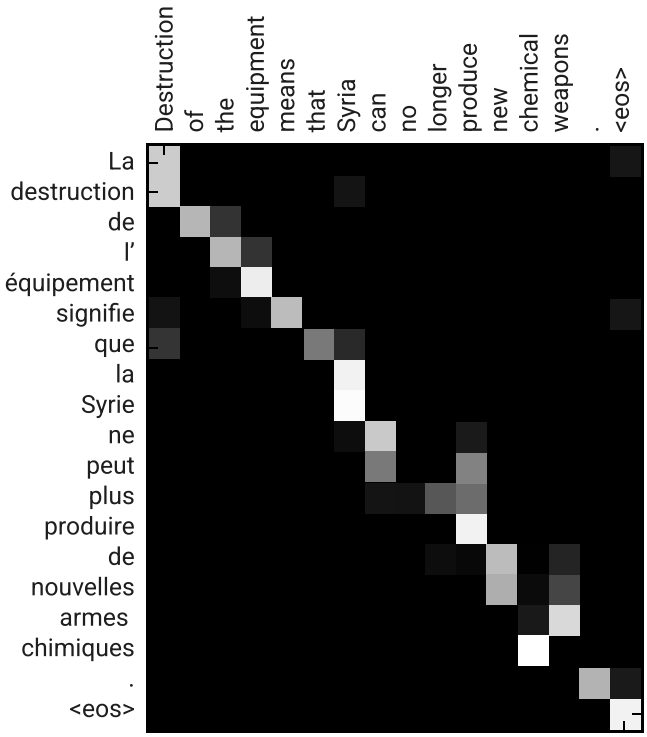


Figure 2.5: Attention matrix example showing the alignments between the French sentence predicted by an NMT system (y-axis) given the English sentence (x-axis). Reproduced from Bahdanau et al. (2015).

Self-Attention Another type of attention mechanism, self-attention (used by the Transformer architecture), contrary to sequential source-target attention, allows processing all the (either source or target) sentence’s tokens at once and model contextualized relations between each sentence’s words, as seen in Figure 2.6. Another particularity of the self-attention approach proposed in Vaswani et al. (2017), is the multi-head mechanism, in which several self-attention

matrices are computed independently, which are marked in Figure 2.6 as the 8 different colors, which stand for the 8 attention heads featured in the vanilla `Transformer Base` architecture. The intuition behind this is that each one of the heads will be able to capture different relations between the sentence tokens.

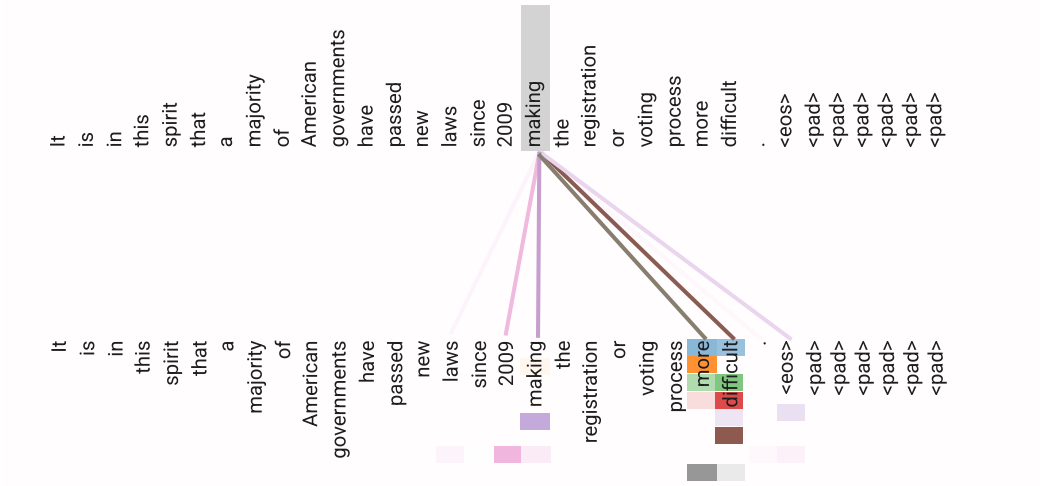


Figure 2.6: Self-attention matrix example as produced by the `Transformer` model. It shows the dependencies between tokens of the same sentence that are useful to produce the current output token. The attention block colors stand for each one of the 8 multi-head attention mechanisms (here aligned horizontally). Reproduced from Vaswani et al. (2017).

2.4.2 Evaluation metrics

`(Sacre)BLEU` As a corpus-level automatic translation quality metric, we systematically report results using case-sensitive BLEU (Papineni et al. 2002), as it is the standard way to quantify translation performance of the MT systems. More specifically, we consistently report results using its tokenization-agnostic versions, that is, `SacreBLEU` (Post 2018) and `multi-bleu-detok.perl` [1], both taking the detokenized translation hypothesis as input and performing their own internal tokenization, which can be easily reproduced. This follows the recent trend of evaluating translations without any specific tokenization (other than space delimiters between words) to ensure the reproducibility of the evaluation regardless of the different tokenization rules that different researchers may use. Indeed, the necessity to report results using detokenized evaluation frameworks has been widely acknowledged (Marie et al. 2021) as this BLEU metric flavor is increasingly popular over traditional BLEU score.

[1] https://github.com/moses-smt/mosesdecoder

Edit distance Metrics based on BLEU are evaluating translation quality at the corpus-level: they are not accurate when ranking single sentences or small corpora due to the high sparseness of n-grams in these conditions (Ye et al. 2007). In consequence, correlation between human evaluation and sentence-level BLEU can be astonishingly low (Liu and Gildea 2005; Li et al. 2016). In order to have a sentence-level metric, which enables single translation outputs to be ranked and compared, we have chosen to use token-level edit distance between predictions and references (Przybocki et al. 2006). Having a sentence-level metric proves useful in Chapter 6 for a fine-grained characterization of UGC and the challenges it raises.

chrF The BLEU metric has the limitation of needing an exact token-level match between prediction and reference, thus, a change in the lexicon or relatively small and fine-grained errors (e.g. a predicted token differing in a single character to that of the reference translation) are not quantifiable. Hence, we also use the MT character-level `chrF` (Popović 2015) evaluation metric, which we use to compare to BLEU metrics and explain UGC translation quality taking into account character-level editions. This metric proved to be the most reliable when characterizing the robustness of MT to UGC in Chapter 6 and Chapter 7.

`chrF` is defined as the *F-score* on character n-grams that overlap between the hypothesis and reference:

$$chrF\beta = (1+\beta^2)\frac{chrP \cdot chrR}{\beta^2 \cdot chrP + chrR} \tag{2.7}$$

where $chrP$, the character n-gram precision, is the percentage of n-grams in the hypothesis contained in the reference; and $chrR$, the character n-gram recall, is the percentage of n-grams in the reference also present in the hypothesis. β is a parameter that gives proportionally more importance to recall than to precision. In this work, we have chosen $\beta = 1$, which gives the same importance to the recall and precision.

2.4.3 Assessing the performance impact of noise in the source sentence

We are particularly interested in assessing the translation quality impact caused by noise by the means of comparing the performance achieved by a given MT system when translating "raw" noisy sentences to the translation output by the very same system when translating the "normalized" version of the same sentences (i.e. sentences with the same meaning in which the

noise has been "removed"). As notation to further discussing this kind of metrics, we will refer to x as the original noisy source, $\tilde{x}$ as its normalized version, and y, $\tilde{y}$ their respective translation hypothesis.

It should be noted that these metrics require "normalized" corpora, in which noisy source sentences have been "corrected" to a form that complies with the language's rules. Only a few normalized corpora are freely available, as this kind of annotation is difficult to accomplish because of potential ambiguity in the source text. This problem is all the more important because it is often necessary to produce new references corresponding to the expected translations when the sources do not contain noise, since removing it from the sentences can also remove information and thus modify the corresponding reference translation (e.g. if emojis or emoticons are removed from the input, they should also be suppressed in the reference). These considerations raise a major resource-intensive limitation that makes such metrics costly and renders annotation consensus notably hard. In addition, this kind of annotated UGC corpora is highly dependent on the specificities and scheme used. For instance, Fujii et al. (2020) provides a UGC corpus annotated in terms of four static UGC specificities, but it remains difficult to extend it to other noise categories.

Throughout this work, we use two metrics to assess the robustness of each NMT model by quantifying to what extent the noise in the source impacts translation quality. They are based on a monotonically increasing metric with respect to the translations' quality, and which we proceed to elaborate on using the BLEU score; although they can be adapted to be used for the-lower-the-better metrics trivially.

Noise impact ratio (NR) The noise impact ratio (Fujii et al. 2020) is the quotient between the BLEU score obtained by translating the original, noisy sentences evaluated using the original references, and the BLEU score achieved by inputting the normalized version of the same source sentences instead and evaluating using the accordingly normalized references. It is calculated as:

$$NR(\boldsymbol{x}, \tilde{\boldsymbol{x}}, \boldsymbol{y}, \tilde{\boldsymbol{y}}) = \frac{\text{BLEU}(\boldsymbol{x}, \boldsymbol{y})}{\text{BLEU}(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}})} \tag{2.8}$$

In other words, this metric compares the translation quality achieved by an MT system when

translating noisy sentences to the one that would have been obtained if no noise had been present in the source. Hence, an NR score of 1.0 means that the noise in the source sentence does not impact the MT quality at all (i.e. the MT system was able to automatically remove the noise present in the source sentence or least ignore it); conversely, the smaller the score, the greater the impact that noise has on translation quality.

Target-source noise ratio (TSNR) Proposed by Anastasopoulos (2019), this metric aims to assess how much perturbation is artificially added during the MT system due to the noise initially present in the input sequence. It is defined as:

$$TSNR(\boldsymbol{x}, \tilde{\boldsymbol{x}}, \boldsymbol{y}, \tilde{\boldsymbol{y}}) = \frac{100 - \text{BLEU}(\boldsymbol{y}, \tilde{\boldsymbol{y}})}{100 - \text{BLEU}(\boldsymbol{x}, \tilde{\boldsymbol{x}})} \tag{2.9}$$

This is to say, by using this metric, we quantify to what extent the noise present in the input sentence (estimated by BLEU $(\boldsymbol{x}, \tilde{\boldsymbol{x}})$) is compared to the noise output by the system when producing the target translation (estimated by BLEU $(\boldsymbol{y}, \tilde{\boldsymbol{y}})$. Concisely, a TSNR value close to 1 indicates that the noisy textual perturbations (assessed using BLEU) in the input are comparable to those of the resulting translation. This is to say that the drift UGC specificities present in the source On the other hand, a low TSNR value suggests that the MT system is less prone to propagate the source-side noise and, consequently, generating a relatively more similar translation to the translation of the normalized version.

2.5 Conclusions

In this chapter, we have given a broad vision of the methods used in this thesis and how they compare, which commonly translates into advantages and disadvantages regarding memory size, computation time, and models' complexity.

We have discussed the main MT model families, focusing on the different NMT architectures we address in the next chapters. In addition, we have presented how we exploit the information provided by the attention matrices for the error analysis when exploring how different MT technologies behave when translating UGC in Chapter 4.

In the final part of this chapter, we presented and discussed the metrics we use in the next

chapters to evaluate the translation quality of the explored MT systems. Especially, we reviewed *TSNR* and *NR*, that we use to assess MT robustness to UGC. We also discussed challenges regarding the use of such metrics, which require additional annotation efforts to normalize UGC, ultimately leading to complex normalization guidelines and difficult inter-annotator consensus.

Now that we have established our scientific framework and explained the set of MT methods that will be used throughout this work, we proceed to describe and discuss UGC and elaborate on the challenges and the interests it poses for NLP, specifically MT.

Chapter 3

User-generated content: an NLP nightmare

In this chapter of the dissertation, we introduce in further detail the main task to be addressed in this work and why it is as problematic as it can be useful to conceive robust NLP models that can cope with a wide range of linguistic phenomena. As we will discuss below, UGC is widely and increasingly present in today's discussion channels and can provide an interesting source of data, although it is characterized by extremely high lexical and semantic variability.

3.1 What is UGC and what is at stake?

UGC consists of different types of content, such as pictures, videos, audio, and texts created and published on the Internet by its users (George and Scerri 2007). Concretely, under the scope of this work, we address text-based forms of UGC, specifically, noisy social-media UGC, or non-canonical text, which is a kind of written content that is produced by users in online discussion channels such as `Twitter`, `Facebook` or `Doctissimo`. This type of texts has been extensively studied over the years, *e.g.* (Foster 2010; Seddah et al. 2012; Eisenstein 2013; Sanguinetti et al. 2020). In this context, users are often able to add elements that are typically not presented in canonical corpora in order to denote emotions, shortening typing time or efforts and, thus, promoting spelling, grammatical, and semantic diversity, whether purposely or not. Despite the challenges posed by UGC, which we will discuss in this chapter, such content can provide a direct way to estimate customer needs and trends (Bahtar and Muda

2016; Timoshenko and Hauser 2019) and thus can represent a powerful asset for data mining due to their ubiquitous presence on the Internet (Krumm et al. 2009; O'Hern and Kahle 2013).

As an easily available source of data, UGC has also provided a interesting, if not valuable, source of data for different NLP tasks, such as hate speech detection (Mossie 2020; Gomez et al. 2020), prevention against human trafficking (Chambers et al. 2019), sentiment analysis (Schmunk et al. 2013), fact-checking (Bondielli and Marcelloni 2019) and mental health assessment (Calvo et al. 2017).

Almost by definition, UGC covers multiple different domains; is multi-modal in nature (Khoshamooz and Taleai 2017), addresses diverse topics and because of the diversity of the users, entails a large degree of language variation (Sanguinetti et al. 2020). Interestingly, these variations are perceptible also through many orthographic variants that span over named-entities (Jijkoun et al. 2008). Specific idioms and domain-related jargon make clear the fact that UGC does not constitute an homogeneous domain and instead could be seen as a multidimensional space where each axis would relate to one specific aspect (domain, divergence to the orthographic/grammatical norm, sentiment expression, socio-demographic characterization of the user, etc.) (Foster 2010; Seddah et al. 2012; Eisenstein 2013; Martínez Alonso et al. 2016).

Exploiting this type of text and, therefore, designing translation systems capable of treating them has been an crucial ongoing research question in the literature (Chen 2022). As coping with the noise that rifles UGC still poses many challenges[1], the questions about the extent to which an UGC phrase effectively conveys information remains, as opposed to being gibberish or the cause of an unfortunate mistranslation.

For example, the automatic translation of a noisy Arabic version of "good morning" triggered a terrorist alert when Facebook mistranslated it, resulting in an unjustified arrest in 2017 (Guardian 2017). A similar unfortunate situation took place massively when the `Facebook` translation engine translated thousands of Burmese posts that contained the name of the Chinese president, Xi Jinping, into a slur word in English, forcing the company to issue a public apology (Times 2020). Beyond these scandals, modern NLP pipelines continue to make mistakes when faced with noisy input, and proposing architectures that overcome these issues is still an open question.

[1] Cf. the long standing series of workshops on Noisy User-Generated Text http://noisy-text.github.io/2022/.

Encoding simplification: covers a set of phenomena aiming to reduce writing efforts		(English trans.)
Word omission	"je n'aime pas" → "j'aime pas"	I don't like
Irregular tokenization	"I was" → "Iwuz" (contraction)	
	"c'était" → "c t" (over-splitting)	It was
Spoken language	"je ne sais pas" → "chais pa"	I don't know
Marks of expressiveness: denote emotion and expressiveness of the author		
Emojis and emoticons	:)	
Punctuation reduplication and Graphemic stretching	"Troooooooppp !!!!" → "Trop !"	Too much
Context dependency: encompasses all the tokens related to the ongoing discussion		
Proper nouns or NE	*Flappy Bird*	
Hashtags and user mentions	#CoupeDuMonde, Jonhdoe11	
Spelling and typographical errors	"Je fsit" → "Je fais"	I do
Grammatical errors	"There are a lots", "Elle es grand"	
Jargon	"mec", "wesh"	Dude/man
Profanities (occasionally masked)		
Dialects		
Code-switching	"C'est trop good"	It's too good
Phonetic writing errors	"J'ai regarder" → "J'ai regardé"	I watched
Internet slang	"to be honest" → "tbh"	
Inconsistent casing	"Allez !" → "ALLEZ !"	Go

Table 3.1: Examples showing common noisy phenomena in UGC (Seddah et al. 2012).

3.2 Comparing UGC to canonical texts

UGC has many specific characteristics that explain why it differs from canonical texts (Sanguinetti et al. 2020; Michel and Neubig 2018), which are, in contrast, clean and well-formed corpora that comply with the set of linguistic rules of a given language.

Regarding the different noise specificities that are prevalent in UGC, we display, in Table 3.1, the ones we consider and discuss in further detail in Section 3.3.

Because of these specificities, UGC text differs from widely distributed canonical corpora, used to train reproducible (and usually license-friendly) NLP models. These differences, which, among others, result in a high number of Out-of-Vocabulary (OOV) tokens, raise many challenges to the automatic processing of UGC. First, large UGC corpora are expensive to annotate and can intrinsically impose major consensus issues due to ambiguities and, occasionally, incomprehensible phrases. Second, due to the extremely high productivity of UGC, which permeates this type of text at all linguistic levels (lexical, grammatical, and syntactic); there are too many morphological variations of every possible linguistic construct to account for all of them. Specifically, MT, which constitutes the main scope of this work, illustrates the difficulties of processing UGC, as performances are affected when translating due to its specificities.

In this regard, we now present different lexical, grammatical, and syntactical UGC speci-

ficities that make it different from canonical corpora in terms of a well-defined linguistic and morphological classification. Thus, in order to illustrate the difficulties imposed by translating UGC, we show some examples and their translations by public and popular translation systems in Table 3.2. Indeed, a decrease in translation quality has been documented when MT systems have to translate this kind of text (Khayrallah and Koehn 2018; Fujii et al. 2020). In this respect, we use the phenomena discussed in the following to explain and identify the caveats of MT when facing UGC in terms of its distinctive linguistic traits.

Cr#pbank	src	ma **TL** se vide après **j'me fais chier** puis **jme** sens seule mais **c** surtout **pdt les vacs mais c pas le cas dc** ça compte pas trop
	ref	my TL is emptying then I get bored then I feel alone but mostly during holidays but it's not the case so it's not so important
	google	my TL is empty after I'm pissing then I feel alone but c especially pdt vacs but it does not matter that it does not count too much
	src	Dans Flappy Bird quand mon Bird il fait 10 jsuis trop contente #TeamNul Mdddr
	ref	At Flappy Bird when my Bird has 10 Im so happy #TeamFail Loool
	google	In Flappy Bird when my Bird is 10, I'm too happy #TeamNul Mdddr
	src	**Booooooooooooooooooooonnne Annniversaure Ma viiiiiiiiiiiie jtm** plus que tout profite bien de tes **22ans moaaaaaaaaa**
	ref	Haaaaaaaaaaaaaaaaaaaaapppy Biiirthday My liiiiiiiiiiiiife luv you more than anything enjoy your 22years mwaaaaaaaaah
	google	Booooooooooooooooooooonnne Annniversaure My viiiiiiiiiiiie jtm more than anything enjoy your 22 years moaaaaaaaaa
MTNT	src	**AJA** que le XV de France féminin est **un train de faire** un grand **chelem, OKLM**
	ref	TIL that the XV of Feminine France is doing a grand chelem, FRESH
	google	AJA that the XV of France feminine is a train of a grand slam, OKLM
	src	Je pensais mais c'est le même **ident et mdp** que mon compte "normal", et il détecte même la **profile pic** et le nom
	ref	I thought so but it's the same username and password as my 'normal' account, and it detects even the profile pic and the name
	google	I thought but it's the same ident and mdp as my "normal" account, and it even detects the pic profile and the name
	src	**Aaaaaaaah.... 8 ans après, je viens de percuter.... :o 'tai** je me disais bien que je passais à côté d'un truc vu les **upvotes.**
	ref	Aaaaaaaah.... 8 years later, I've just realized.... :o damn I had the feeling that I was missing something considering the upvotes.
	google	Aaaaaaaah .... 8 years later, I just collided ....: oh well I was telling myself that I was missing something seen the upvotes.

Table 3.2: Examples from both UGC showing the source phrase, reference translation, and Google Translate output. UGC specificities are highlighted using bold characters. *Translation site accessed on 28-01-2021.*

3.3 UGC specificities

In this section, we proceed to discuss in detail the UGC's idiosyncrasies previously introduced. In order to characterize UGC, we consider a list of 13 syntactical, grammatical, and morphological UGC specificities presented in Table 3.3, which is based on the typology of Sanguinetti et al. (2020). These categories represent distinct lexical and grammatical variability that can be

UGC's specificities
Encoding simplification • Letter change/deletion/addition • Diacritic omission/error • Phonetization • Contraction over-splitting • Wrong verb tense • Wrong gender/grammatical number • Abbreviation
Marks of expressiveness • Inconsistent casing • Emoticons/smileys • Punctuation reduplication and Graphemic stretching • Interjections
Boundary shifting • Tokenization errors
UGC Context-dependent • UGC-specific characters • Named Entity

Table 3.3: Typology of UGC specificities used in this work and as our manual annotation scheme. Extended from Sanguinetti et al. (2020).

classified and which we investigate in terms of their impact on translation quality. Our defined types differ from the original typology only by not considering code-switching and profanities but, on the other hand, adding a category for interjections.

To introduce the considered typology, we will start by explaining the main groups of UGC's specificities in Table 3.1, which in turn we divide in the fine-grained typology presented for the experiments in this thesis. Essentially, these groups are classified in a purpose-oriented manner, i.e. they encompass the lexical, grammatical, and syntactical peculiarities, characterized in terms of concrete morphosyntactic specificities of user expression while associating a reason for employing such linguistic resources. For instance, in UGC, users often simplify the message to reduce writing efforts or express emotions more vividly, which often leads to a very high lexical variability, such as those seen for the "Word omission" and "Punctuation reduplication" categories in Table 3.1.

We also base some of the UGC specificities on a context and domain-related axis (Martínez Alonso et al. 2016), for example, Named Entities (NE) evoking a certain television series.

3.3.1 Encoding simplification

This kind of UGC's phenomenon involves a large set of noisy occurrences, commonly used by users for shortening the messages' length, while enjoying the freedom to introduce new morphological variations for otherwise known and comprehensible words. Arguably, they are responsible for a large number of *spelling errors*, consequently introducing OOVs and rare token sequences. In Table 3.4, we present artificial examples of how these UGC specificities could affect the quality of the translation individually, which we will comment on in more detail.

Letter change, deletion or addition Words that present some form of letter alternation or modification from the correct form, introduced by the user willingly or not. They often lead to OOV tokens and some can compromise the full meaning of the translation, e.g. "*Je suis alé à la bibliothèque*" is translated from French to English by Google Translate as "*I am random at the library*" in example ①. The error appears to be that the MT system predicted *alé* as *aléatoire*, which is translated as *random* in English, whereas the correct spelling should have been *allé* (*went* in English). This encoding simplification phenomenon can be seen as the omission of the missing letter (*l*) to match the correct French word.

Diacritics omission or error As in character-level modifications for an existing word, diacritic errors often cause OOVs, but the size of the search space of possible combinations is substantially reduced by the fact that each letter can only be changed by the same letter with different diacritics (i.e. accents). State-of-the-art MT models later proved remarkable robustness to these UGC specificities further in this work, and we found no example containing only this type that produces an incorrect translation in GOOGLE TRANSLATE when consulted.

Encoding simplification			
	① Letter change/deletion/addition		② Abbreviation
SRC	Je suis **alé**[allé] à la bibliothèque	SRC	Merci **bn**[bien], les amis.
REF	I **went** to the library	REF	Thank you **very much**, friends.
Google	I am **random** at the library	Google	Thank you **bn**, friends.
	③ Phonetization/spelling mistake		④ Wrong verb tense
SRC	Il est **nez**[né] pour **sa**[ça].	SRC	Je suis **allez**[allé] à l'école.
REF	He was **born** for **it**.	REF	I **went** to school.
Google	He is nose for his	Google	I **am going** to school

Table 3.4: Examples of translation impact due to Encoding Simplification phenomena. The references (REF) are translations by the same MT engine of the correct form, accordingly. Translations were produced on the 07/03/2022 *Normalized version of the tokens is displayed in blue brackets.*

Phonetization It takes place when a word is written using an alternative orthography (existing or not) that shares a similar pronunciation. This phenomenon involves knowing the pronunciation to match it with its corresponding canonical form. For instance, in example ③ in Figure 3.4, "*nez*" is translated using its literal meaning to "*nose*" in English, whereas the correct source French token is "*né*", sharing the same pronunciation /ne/. The same happens for the tokens "*sa*" et "*ça*", leading to substantial changes in translation, as can be seen.

Wrong verb tense We study grammar conjugation errors that can lead to ambiguities and could cause errors in translations.

Wrong gender or grammatical number We also take into account grammar inconsistencies to explore whether they represent an inconvenience to MT performance and a source of ambiguity during translation. These are annotated with respect to grammatical differences in gender and number.

Abbreviation These constitute a rather high compression modification of the message, including the omission of letters for any given word and compacting multi-word expressions (MWE) in a reduced sequence of characters. They most often lead to OOV tokens that could be mapped to a sequence of known target tokens, although it often proves challenging since it requires, to a large extent, a lexicon. In example ⑥, the abbreviation "*bien*" → "*bn*" for a fairly standard French expression fails to produce the correct translation ("*very much*").

3.3.2 Marks of expressiveness

These account for any modification or addition of words, letters and punctuation symbols in the message in order to convey emotions. In this work, we considered three specific UGC morphological features that are related to such a group. Examples of the impact on translation quality are shown in Table 3.5, referenced in this subsection.

Emoticons and emojis The former corresponds to sequences of characters to state the intended tone and denote emotions (often by representing facial expressions using punctuation symbols) (Dictionary 2017), and consequently introduces rare sequences of often known source

tokens. On the other hand, emojis, are specially encoded characters that represent miniaturized figures, serving, in a general way, to the same purposes as emoticons, but introducing an ever-increasing list of valid emojis (3,251 emojis in Unicode Standard, as of September 2020). Although emojis are progressively present in today's linguistic landscape (the 2015 Oxford Language word was, indeed, an emoji (Verge 2015)), their high number and variability often results in character-level OOV during evaluation. In example ①, the Unicode-encoded emoji ("*U-1F972*"), a character-level OOV, is ignored during translation. Conversely, the emoticon "*:O)*", representing a surprised expression, being a rare sequence not present in canonical texts, is treated by the MT engine as "***o**ù*", producing its corresponding English translation "*where*".

Marks of expressiveness			
	① Emoticons and emojis		② Grapheme stretching and Punct. reduplication
SRC	C'est vrai **:O** il est là **U-1F972**.	SRC	C'est **supeeeerrrr !!!**[super!].
REF	It's true **:O** he is **there** **U-1F972**.	REF	It's **great!!!**
Google	It's true: **where** he is	Google	It is **supereeeerrrr !!!**
	③ Inconsistent casing		④ Interjections
SRC	Ça c'EsT TRÈS BiEn.	SRC	C'est ça **ptdr**.
REF	That IS VERY GOOD.	REF	That's it, **lol**.
Google	THAT IS VERY GOOD.	Google	This is **ptdr**.

Table 3.5: Examples of translation impact due to Marks of Expressiveness phenomena. The references (REF) are manual edits of Google's translations of the correct form to include the missing characters. *Normalized version of the tokens is displayed in blue brackets.*

Grapheme stretching and Punctuation reduplication They consist of repetitive structures to emphasize emotion by expanding certain words or punctuation. These are most often manifested in rare repetition sequences of characters and punctuation marks. Example ② shows how these specificities can affect translation by adding repeated characters. Punctuation repetitions ("*!!!!*") are arguably easier to treat, as they can be tokenized and recopied in the output, while words undergoing stretching have to be mapped to a known word and translated accordingly ("*supeeeerrrr*" in the example).

Inconsistent casing This UGC specificity involves any kind of casing occurrence that does not comply with the language rules. They introduce different (possibly alternated) upper- or lowercase versions of characters or full words, which, in turn, are represented by completely different elements in the vocabulary. However, current MT systems employ methods such as translating the full lowercase version of the text and employing placeholders to recover the actual

source-side casing, known as in-line casing (Berard et al. 2019b). The public translation engine appears to be robust to this UGC specificity, as can be seen in example ③. However, it is an additional confounding factor for translation; for instance, in the presence of a letter deletion, Google's translation system outputs: "*C'est très bin*" → "*It's very good*", whereas, when an inconsistent case is introduced, the result is "*C'est très BIN*" → "*It's very BIN*".

Interjections These are words often related to MWE that also set the intended tone of the message. They are specifically difficult to process correctly without any lexicon or example dataset, since they represent rare sequences that can be extended into spans of several words. In example ④, the interjection "*ptdr*", which stands for "*peté de rire*", which is analogous to *lol* (*laughing out loud*) in English, is not recognized by the translator and is copied into the output. Some interjections are correctly translated by Google's engine, e.g. it is the case if we substitute "*ptdr*" by "*mdr*", another known French to evoke laughter, which is translated as "*lol*".

3.3.3 Boundary shifting

This category groups the phenomenon of changing the segmentation between words, either to reduce writing efforts or due to typographical errors.

Tokenization errors We include in this group any sequence of words whose segmentation (mostly delimited by *spaces* and apostrophes in French and English) is not in agreement with its canonical segmented form. In the examples in Table 3.6, we can see how these UGC's specificities can impact translation.

Boundary shifting			
	① Contraction		② Over-splitting
SRC	**Jviens**[Je viens] **defaire**[de faire] un bon marché.	SRC	C'est **p lutôt**[plutôt] pas mal ça.
REF	I **just did** a good **deal**.	REF	It's **pretty good**.
Google	I **have come** to do a good **market**.	Google	It's **not bad soon**.

Table 3.6: Examples of translation impact due to Boundary Shifting phenomena. The references (REF) are translations by the same MT engine of the correct form, accordingly. *Normalized version of the tokens is displayed in blue brackets.*

3.3.4 UGC context-dependent specificities

These make reference to any peculiarities of the message related to the specific nature of the channel or the exchange platform used. In this work, we divide them into two types, with corresponding examples in Table 3.7:

UGC Context-dependent specificities			
	① Special characters		② Named entities
SRC	#Tenez bon ça va durer.	SRC	c'est excellent de jouer à **fish fins**.
REF	#Hold on it's going to last.	REF	It's excellent to play **fish fins**.
Google	#Get good it will last.	Google	It is excellent to play **fine fish**.

Table 3.7: Examples of translation impact due to Marks of Expressiveness phenomena. The references (REF) are manual edits of Google's translations of the correct form to include the missing tokens. *Normalized version of the tokens is displayed in blue brackets.*

Special character sequences These are occurrences of characters that, due to the UGC's extreme diversity could be, in principle, any type of character, possibly in a different encoding. They can also depend on the UGC platform having distinguishable uses to access functionality, e.g. the characters *#* and for hashtags and user mentions, respectively, in TWITTER or URL strings (*http(s)://*). In example ①, the character *#* prevented the correct translation of "*tenez bon*" $\rightarrow$ "*hold on*".

Named entities They are NE that are in line with the informational context surrounding the UGC, e.g. names of films or games (possibly written in a foreign language). Some other NE depend on the publication platform, as is the case of *RT* and *TT*, denoting *Retweet* and *Trending Topics* in TWITTER discussions. In example ②, the name of a hypothetical game (*fish fins*) is incorrectly translated when it should be kept unmodified in the output.

3.4 UGC as an ever-changing way of expression

In addition, motivated by the observation that the potential variations in UGC are too wide and variable to be accounted for, and, lacking sufficiently large translated UGC training datasets in open access, we set as our main approach not to use any domain-adaptation technique nor target-specific (UGC) data. Furthermore, a constant evolution of ways of expression, such as the increasing use of an ever-growing set of available emojis and endless possible discussion

topics, gives UGC a considerable dynamic through time and location of the users (Broni 2021), and that has recently been shown to strongly contribute to emergent linguistic constructs (Raviv et al. 2020).

Furthermore, Oren et al. (2019) highlighted the disadvantages of fine-tuning, which requires knowing the test distribution, especially variable for UGC; and training a different model or module that specializes in targeting a domain distribution. These limitations ultimately pose an overhead for MT systems and require *a priori* understanding of the test distribution. The authors also assessed that using text outside the main target distribution, such as in UGC-augmented training corpora, actually degrades performance on the original target distribution (canonical text used during training).

In this regard, we focus our research on exploring methods that can generalize over a wide range of noisy phenomena when testing, which, in turn, aim to produce more robust MT systems without using any specific UGC text. Concretely, we have systematically chosen training corpora constrained to canonical texts and zero-shot evaluation of UGC test sets, these being considerably outside the distribution of our training datasets, as we will quantify and discuss in further detail in Section 3.6 At this point, it is worth noting that, since we want to conceive MT architectures that can adapt and generalize over new forms and noisy variations of the canonical training datasets, we are particularly interested in Neural Machine Translation (NMT) approaches. In this train of thought, our work mainly focuses on proposing methods that can enhance neural learning representations of noisy constructs, either by pre-processing pipelines or architecture modifications that translate into performance improvement when processing UGC.

In this zero-shot scenario, we intend to resolve UGC specificities without using any UGC data, for which we explore and propose methods that leverage some heuristics or fine-grained morphological properties. Under these premises, we proceed to state the challenges and main research questions for this thesis work.

3.5 Datasets

In this section, we describe the datasets for training, developing, and testing all of the MT models investigated throughout this work. We elaborate both on the canonical and commonly used

corpora for MT, as well as the UGC ones, which we compare in terms of automatic metrics for noise and domain-drift assessment (discussed in Section 2.4.2).

Canonical data We train our models on two different canonical parallel corpora. We first consider the traditional corpus for training MT systems, namely the WMT data consisting of the `europarl` (v7) corpus[2] and the `newscommentaries` (v10) corpus[3]. We use the `newsdiscussdev2015` corpus as a development set. This is exactly the data confgiuration used to train the system described in Michel and Neubig (2018), which will be used as a reference throughout this work.

We also consider the French-English parallel portion of `OpenSubtitles'18` (Lison et al. 2018) as a second training set. This corpus is a collection of crowd-sourced and peer-reviewed edited subtitles for movies. We assume that because it is made up of informal dialogues, such as those found in popular *sitcoms*, sentences from `OpenSubtitles` will be much more similar to the UGC data than the `WMT` training dataset. However, it must be noted that UGC differs significantly from subtitles in many aspects: emotion denoted with repetitions, typographical and spelling errors, emojis, etc.

To allow for a fair comparison between systems trained on `WMT` and on `OpenSubtitles`, we consider a `small` version of the `OpenSubtitles` corpus, that has nearly the same number of tokens as the `WMT` training set and a `large` version that contains all `OpenSubtitles` parallel data.

To evaluate our system on in-domain data, we use two test sets, namely, `newstest'14`, as well as 11,000 sentences extracted from `OpenSubtitles`, which we refer to as `OpenSubTest` throughout this dissertation. Statistics for our different datasets are presented in Table 3.9.

Non-canonical UGC data To evaluate our models, we consider two data sets of manually translated UGC.

The first is a collection of French-English parallel sentences manually translated from an extension of the Parallel French Social Media Bank (Seddah et al. 2012), which contains texts collected on Facebook, Twitter, as well as from the forums of `JeuxVideos.com` and

[2] www.statmt.org/europarl/
[3] www.statmt.org/wmt15/training-parallel-nc-v10.tgz

Doctissimo.fr.[4]

We refer to this corpus as PFSMB for Parallel French Social Media Bank, and it consists of 1,554 comments in French annotated with different kinds of linguistic information: Part-of-Speech tags and surface syntactic representations. Phrases have been translated from French to English by a native French speaker and an extremely fluent, near-native English speaker. Typographic and grammatical errors were corrected in the gold translations, but the linguistic register was kept. For instance, idiomatic expressions were mapped directly to the corresponding ones in English (e.g. "mdr" has been translated to "lol" and letter repetitions were also kept (e.g. "ouiii" has been translated to "yesss"). For our experiments, we divided the PFSMB into a test set and a blind test set containing 777 comments each.

We also consider, for these experiments, the MTNT corpus (Michel and Neubig 2018), a dataset made up of French sentences that were collected from Reddit and translated into English by professional translators. We use their designated test set and add a blind test set of 599 sentences we sampled from the MTNT validation set. The PFSMB and MTNT corpora both differ in the domains they consider, their collection date, and the way sentences were collected to ensure that they are sufficiently noisy. Several statistics of these two corpora are reported in Table 3.9. As expected, our two UGC test sets have a substantially higher token-to-type ratio (TTR) than the canonical test corpora, indicating greater lexical diversity.

As discussed previously, we have divided the UGC test sets into two subsets: common test sets and blind test sets. This is motivated by the necessity of unforeseen test corpora in order to fairly report results of our methods after choosing the best configurations and MT systems over the common UGC tests. To gain more insights into the sizes and basic statistics of our entire dataset collection, please refer to Table 3.9, where the statistics of the mentioned corpora are shown.

Some examples of samples from the PFSMB and MTNT UGC corpora, and their corresponding reference translation, can be found in Table 3.8. On the other hand, to evaluate whether our methods keep their generalization properties and, thus, are still able to perform well when processing clean test sets, our evaluation policy is to always display the performance of our MT systems over two canonical test sets: newstest'14 and OpenSubTest. Both of these test corpora cover the domain of our training data respectively: formal linguistic forms featured in

[4]Popular French websites devoted respectively to video-games and health.

UGC Corpus		Example
MTNT	FR (src)	Je sais mais au final c'est moi que le client va supplier pour son offre et comme **Jsui** un gars cool, **jfai** au mieux.
	EN(ref)	I don't know but in the end I am the one who will have to deal with the customer begging for his offer and because **I'm** a cool guy, **I do** whatever I can to help him.
PFSMB	FR (src)	si vous me comprenez **vivé** la **mm** chose ou *[vous]* **avez passé le cap** je **pren tou** ce qui peu m'aider.
	EN (ref)	if you understand me **leave** the **same** thing or **have gotten over it** I **take everything** that can help me.

Table 3.8: Examples for our considered UGC corpora. Noisy tokens and their corresponding translations are shown in bold font.

WMT and considerably more colloquial ways of expression in OpenSubtitles.

Corpus	#sentences	#tokens	ASL	TTR	#chars
train set					
WMT	2.2M	64.2M	29.7	0.20	335
OpenSubtitles	9.2M	57.7M	6.73	0.18	428
test set					
OpenSubTest	11,000	66,148	6.01	0.23	111
newstest'14	3,003	68,155	22.70	0.23	111

Corpus	#sents	#tokens	ASL	TTR	#chars
UGC test					
PFSMB	777	13,680	17.60	0.32	116
MTNT	1,022	20,169	19.70	0.34	122
UGC blind					
PFSMB	777	12,808	16.48	0.37	119
MTNT	599	8,176	13.62	0.38	127

Table 3.9: Statistics on the French side of the corpora used in our experiments. *TTR stands for Type-to-Token Ratio, ASL for average sentence length.*

3.6 Quantifying the difference between canonical texts and UGC

As a first exploratory approach for the corpora being considered, we can note their token-level statistics in Table 3.9. It is worth noting that the TTR of our four UGC test sets are roughly 42% higher than the train and canonical test corpora, which indicates a proportionally higher lexical variability for UGC.

Several metrics have been proposed to quantify domain drift and extraneous n-grams apparitions overall between two corpora. In particular, the perplexity of a language model and the Kullback-Leibler (KL)-divergence between the character-level 3-gram distribution of the train and test sets were two useful measurements capable of estimating the noise-level of UGC corpora as shown respectively by Seddah et al. (2012) and Martínez Alonso et al. (2016). We also employed the perplexity (PPL) of a 5-gram Knesser-Nay language model (Ney et al. 1994) trained on Large OpenSubtitles as a measure to assess the out-of-domain (OOD) (Haddow and Koehn 2012) of a given corpus.

Table 3.10 reports the noise level of our test sets introduced in Section 3.5 with respect to

↓ Metric / Test set →	PFSMB †	MTNT†	Newstest	OpenSubsTest
3-gram KL-Div	1.563	0.471	0.406	0.006
%OOV	12.63	6.78	3.81	0.76
avg. BPEstab	0.018	0.024	0.049	0.13
PPL	599.48	318.24	288.83	62.06

Table 3.10: Domain-related measure on the source side (FR), between Test sets and LARGE OPENSUBTITLES training set. Dags indicate UGC corpora.

our largest training set, LARGE OPENSUBTITLES. The measures show how divergent our UGC corpora are from our largest training set. As shown by its OOV ratio, PPL, and KL-divergence value, the PFSMB corpus is considerably noisier than the MTNT corpus, making it a more difficult target in this translation scenario. We can also notice that both UGC test sets are considerably dissimilar to other canonical OOD set (newstest'14), especially in terms of OOV ratio and BPE piece stability (BPEstab). This metric, proposed by us in Rosales Núñez et al. (2019), quantifies the average lexical diversity of a test set for a given BPE tokenization model, i.e. low average BPE stability points to a more variable BPE neighborhood for the average BPE piece, and thus, higher average vocabulary complexity. Specifically, if N is the total number of BPE tokens in a tokenized corpus, j each of the words of the BPE vocabulary (set to 16K) and $freq_j$ the frequency of such a BPE token j in the test corpus, we computed it as follows:

$$\frac{1}{N} \cdot \sum_{j=1}^{16K} freq_j \cdot \frac{\#unique\_neighbors_j}{\#neighbors_j}$$

This corresponds to the weighted average of the diversity of BPE neighbors for the whole fixed-size vocabulary.

3.7 Scarcity of UGC resources for NLP

As previously discussed, UGC has inherent characteristics, such as addressing any possible kind of domain and raising considerable challenges to overcome for consensus and annotation (Martínez Alonso et al. 2016) due to its extreme variability. This makes their evaluation intrinsically a low-resource scenario in NLP (Meftah 2021), as parallel UGC corpora is scarce in publicly available datasets (Lohar 2020).

In contrast, availability of good quality data in sufficiently large quantities strongly conditions the development and training of NLP models for new domains (Bamman 2017). Recently, some

valuable and peer-reviewed UGC collections have appeared, specifically for MT, for example, the MTNT and 4Square corpora. However, they have primary adaptation and evaluation purposes, containing dozens of thousands of training sentences, substantially less than the usual number of samples needed to train today's NMT models. Additionally, noise present in large and automatically scraped UGC corpora for training purposes, such as the ParaCrawl[5] has shown to negatively impact the quality of NMT (Rikters 2018), which in turn leads to a consequent loss of performance over in-domain corpora, even when UGC is available for training.

3.8 Related works in UGC translation

In this section, we review the works related to the series of MT methods and protocols developed to cope with UGC throughout this dissertation. We also link each of the following sets of MT literature to our proposed approaches in each chapter to justify their relevance at each stage of this work.

3.8.1 MT of UGC: general overview

We start by discussing the literature on the impact on translation performance caused by UGC translation. In this sense, we first review the work that characterizes UGC translation and its performance related to canonical text MT.

Automatic translation of UGC has proven to be a difficult task compared to canonical text MT (Berard et al. 2019a; Fujii et al. 2020), also involving challenges in terms of robustness and optimal vocabulary choice, due to an arbitrarily high variation of terms and orthography, informal language, and spelling errors (Berard et al. 2019b). Its differences, compared to other domain adaptation MT task, lie in the fact that UGC can encompass multiple topics while maintaining increased lexical variation and grammar inconsistencies due to its rich productivity (Michel and Neubig 2018), as previously discussed.

We then review the literature comparing NMT and PB-SMT, highlighting their application for UGC translation, which constitute our first MT baselines in Chapter 4. These are later used to identify and report caveats of the main popular MT paradigms when processing UGC. Bentivogli et al. (2016) showed that NMT predictions need less post-edition processing and present lower

[5]http://statmt.org/paracrawl

error rates than PB-SMT in English → German translation of TED talks. On the same train of thought, Bojar et al. (2016)'s results also indicate that NMT is significantly better than PB-SMT, since less post-editing is needed to output a correct translation. On the other hand, Castilho et al. (2017) reports were somewhat ambiguous, showing that NMT has an edge over PB-SMT in terms of fluency and translation errors, while some results reflect an improvement over PB-SMT in terms of post-editing needs. However, none of the aforementioned works were conducted using experimental setups with a purposely high mismatch between training and test corpora. In this line of research, Haddow and Koehn (2012); Koehn and Knowles (2017) found that NMT performs relatively poorly when translating OOD texts, as well as under low-resource conditions. Dowling et al. (2018) also showed a significant improvement of PB-SMT's BLEU score over NMT on -typically- low-resource English → Irish translation. More recently, Anastasopoulos (2019) conducted experiments to determine how source-side error estimation can be related to different grammatical errors in translations using NMT models.

3.8.2 Normalization and handling ambiguities via DAGs

Further during this dissertation, in Chapter 5, we proposed an automatic normalization pipeline to approach noisy versions of text sequences to canonical constructs, which are ultimately expected to perform better, since the inherent OOD-drift nature of UGC gets minimized. In this order of ideas, several works have focused on using lattices to model uncertain inputs or potential processing errors that occur in the early stage of the translation pipeline. For instance, Su et al. (2017) proposed `lat2seq`, an extension of sequence-to-sequence models (Sutskever et al. 2014), which are capable of encoding several possible input possibilities by conditioning the RNN output to multiple predecessors' paths. More recently, Sperber et al. (2017) introduced a model based on `Tree-LSTMs` (Tai et al. 2015), to correct the output of an Automatic Speech Recognition (ASR) system. On the other hand, Le et al. (2008) use lattices composed of written subword units to improve the recognition rate in ASR.

However, none of the aforementioned works focus on processing noisy UGC corpora, and they do not consider the use of phonetizers and pronunciation similarity to recover correct tokens. They aim to correct known tokens, such that a neural language model chooses the best output when an uncertain input is present (typically words with similar pronunciation from an ASR

output). Instead, our approach calculates the phonetization of the source token and candidates are proposed based on their phonemic similarity to it, whether this original word is an OOV or not, i.e. an alternative phonemic orthography, or incorrectly using another existing word due to their pronunciation resemblance.

On the same trend, Qin et al. (2012) combined several ASR systems to improve the detection of OOVs. More recently, van der Goot and van Noord (2018) achieved state-of-the-art performance on dependency parsing of UGC using lattices.

Closely related to this work, Baranes (2015) explored several normalization techniques on French UGC. In particular, to recover from typographical errors, they considered a rule-based system, `SxPipe` (Sagot and Boullier 2008), which produced lattices encoding the alternative spelling of OOVs and used a language model to select the best correction.

Several works have explored different approaches to normalize noisy UGC in various languages. For instance, Stymne (2011) use Approximate String Matching, an algorithm based on a weighted Levenshtein edit distance to generate lattices containing alternative spelling of OOVs. Wang and Ng (2013) employ a Conditional Random Field and a beam-search decoding approach to address missing punctuation and words in Chinese and English social media text. More recently, Watson et al. (2018) proposed a neural sequence-to-sequence embedding that improves `FastText` (Bojanowski et al. 2017) representations with word-level information, which achieved state-of-the-art on the QALB Arabic normalization task (Mohit et al. 2014).

3.8.3 Character-level MT for UGC

The impact of noise on the `charCNN` and `char2char` models has been evaluated by Belinkov and Bisk (2018) and Ebrahimi et al. (2018) by artificially adding noise to canonical texts (the `TEDTalk` dataset). Their results show that the different character-level models fail to translate even moderately noisy texts when trained on 'clean' data, and that it is necessary to train a model on noisy data (either natural or synthetic) to make it robust. Note that, as discussed in Chapter 3, and as of the date of this dissertation, there is no UGC parallel corpus large enough to train an NMT model, and we must rely on the model's ability to learn, from canonical data only, noise- and error-resistant representations of their input that are robust to the specificities found in UGC. This is why, in this work, we are interested in studying the MT performance in a

zero-shot scenario when translating noisy UGC, as discussed previously in Section 3.4.

To the best of our knowledge, no work has studied the performance of the character-based NMT architectures on an actual UGC scenario with real-world gathered and fine-grained annotated noisy sentences. This sets the main motivation for the character-level MT study in Chapter 6.

3.9 Conclusion

In this chapter, we reviewed substantial reasons to be interested in designing robust MT systems that can correctly translate noisy UGC, especially for public and widespread translation engines. As noticed, the ubiquity of UGC in the modern communication world, and the richness of being massively produced by a large number of users, make them also valuable and to be taken into consideration for different NLP tasks.

We discuss the specificities of UGC and provide a fine-grained typology that can be used to explain translation errors in a black-box query-based fashion to ultimately characterize the difficulties involved in this task. Additionally, examples of translations for each UGC phenomenon we consider are displayed and discussed to illustrate how they impact MT quality, notably when using very popular and public translation engines.

Finally, after stating the motivation and translation problems at hand, we reviewed the literature on UGC translation related to each of the approaches studied in this work.

After reviewing the main methods and techniques for MT and discussing the challenges that UGC poses for automatic translation; in the next chapter, we present our first out-of-the-box MT baselines. This set of experiences has a twofold purpose: it helped us identify and characterize the main translation quality problems caused by UGC on the most used MT approaches; and these baseline results are useful to keep track of the impact of possible methods to cope with UGC investigated throughout this dissertation.

Chapter 4

Machine Translation of Noisy UGC: Translating the Impossible

In this chapter, we describe the main experimental protocols, data and MT systems that we use as baselines and architecture backbone for our methods aiming to enhance the quality of MT systems on UGC. In order to keep consistency and ensure a fair comparison between all the methods presented in this work, we have systematically considered translating French UGC to English and identical training and test corpora. All the datasets and experimental protocols correspond to those of our first publication (Rosales Núñez et al. 2019).[1]

4.1 MT noisy translation with state-of-the-art systems

In this chapter, we start by reviewing and analyzing the behavior of off-the-shelf modern PB-SMT and NMT methods when processing noisy UGC, in order to identify difficulties related to UGC translation. In this respect, we intend to use both automatic metrics and detailed error analysis discussed in Section 2.4.2.

Even if NMT has become the predominant MT approach in recent years (Yang et al. 2020), we decided to include a PB-SMT system in our comparison because, to the best of our knowledge, the robustness of this family of methods has never been evaluated on UGC. Indeed, PB-SMT's translation tables may behave differently to UGC specificities; thus, these may be interesting.

[1]The main goal of this thesis is to improve MT performance when translating noisy UGC. Nevertheless, it is worth noting that we started to explore the impact of UGC on the performance of NLP system with a Part-of-Speech task. This first experience, resulted in a publication at TALN in 2018.

4.2 Zero-shot MT and UGC

As we have discussed in Section 3.7, due to its productivity in terms of new forms, new structures, or new domains, UGC is extremely variable and constantly evolving. Robust UGC MT is expected to cope with an arbitrarily wide range of unknown inputs, as UGC has an evolving nature (Lobato et al. 2011), ultimately resulting in new terms over time (and consequently potential OOV tokens). Furthermore, building a parallel UGC dataset large enough to train MT systems is not possible due to the scarcity of good, consistent and license-friendly real-world UGC data, as discussed in Chapter 3. This is why, in this work, we focus on a zero-shot scenario, with training data constrained to canonical, standard, and publicly available corpora, whereas UGC data are used only to evaluate the robustness of our systems. This choice, motivated by the nature of the data and the task at hand, is particularly restrictive and makes the development of translation systems substantially more difficult. In particular, it strongly constraints the use of all the fine-tuning methods, today, at the heart of many NLP methods (Ramponi and Plank 2020). This experimental configuration was recently used as a new track in the WMT 2020 Robustness Shared Task (Specia et al. 2020).

In short, our aim is to improve the inherent robustness of MT models. Thus, in order to unequivocally account for the improvements to our methods, we remove any impact on performance caused by the means of using UGC data during training.

4.3 First baselines: getting started

To set a first comparison point and assess whether our proposed methods and pipelines result in any performance improvement, we started by training three mainstream off-the-shelf MT architectures using PB-SMT and NMT approaches. For the former, we used the well-known `Moses` framework (Koehn et al. 2007a), and for the latter, we chose the two most commonly used encoder-decoder sequence-to-sequence techniques, i.e. a Bidirectional Long-Short Term Memory Recursive Neural Networks (Bi-LSTM) encoder with attention-based decoding (Luong et al. 2015; Michel and Neubig 2018), and a Transformer architecture that relies on a multi-head attention mechanism (Vaswani et al. 2017). The training, development, and test corpora are those described in Section 3.5.

This first study also aims to provide a comparison and error analysis of PB-SMT and NMT when translating noisy UGC. More precisely, our contributions in this chapter are threefold:

- We compare the performance of PB-SMT and NMT systems when translating either canonical or noisy UGC text;
- we analyze, both quantitatively and qualitatively, several cases in which PB-SMT outperforms NMT on highly noisy UGC, and we discuss the advantages, in terms of robustness, that PB-SMT offers over NMT approaches;
- we explain how these findings support previous observations on the limits of recurrent `seq2seq` (Koehn and Knowles 2017) and `Transformer` (Passban et al. 2021) NMT architectures, by studying cases in which, as opposed to the `PB-SMT` system, the attention mechanism fails to provide a correct translation.

The results presented in this chapter have been published in (Rosales Núñez et al. 2019).

4.3.1 Addressing tokenization: adopting the first mainstream robustness techniques

Following the usual practice used in the literature to address the OOV problem, and as stated in Section 2.3, we have adopted Byte-Pair Encoding (BPE) tokenization (Sennrich et al. 2016). This method has enjoyed extensive literature support, and its principle is aimed at leveraging on subword n-gram unit composition in order to resolve possible OOV. That is, using this tokenization method, the MT systems will decompose words in a sentence into a sequence of the *K* most frequent n-grams occurring in the train set to output a translation, where *K*, the vocabulary size, is the only parameter to be set. It is worth mentioning that, as discussed in Section 2.3, any MT system will automatically replace OOV by a special `<UNK>` token during inference, and, although BPE ensures almost no OOV tokens (Araabi et al. 2022), OOV characters (e.g. emojis) and rare characters (e.g. under-represented alphabets in the train set) are the exceptions.

We have chosen to use a BPE tokenization with a vocabulary size of 16K token, as it is a common parameter value for BPE in literature. We only consider this kind of tokenization since it has been virtually used in all NMT systems and systematically outperforms word-level tokenization (Ding et al. 2019; Yang et al. 2020). Each tokenization model is trained on the corresponding dataset used to train the MT system, as described in Section 3.5.

4.3.2 Results

Table 4.1 reports the results of the different models that we have considered. It is important to note that, as stated in Section 2.3, we have implemented an alignment-based method to substitute any possible <UNK> token into the predictions by copying their corresponding aligned match from the source phrase. Alignments have been predicted by FastAlign (Dyer et al. 2013) trained on the same training data as the corresponding MT system.

	PB-SMT				seq2seq				Transformer			
	PFSMB	MTNT	News	Open	PFSMB	MTNT	News	Open	PFSMB	MTNT	News	Open
WMT	20.5	21.2	22.5†	13.3	17.1	24.0	**29.1**†	16.4	15.4	21.2	27.4†	16.3
Small	28.9	27.3	20.4	26.1†	26.1	28.5	24.5	28.2†	27.5	28.3	26.7	31.4†
Large	**30.0**	**28.6**	22.3	27.4†	21.8	22.8	17.3	28.5†	26.9	28.3	26.6	**31.5**†

Table 4.1: BLEU score results for our three models for the different train-test combinations. The best result for each test set is marked in bold, best result for each system (row-wise) in blue, and score for in-domain test sets with a dag. 'News' and 'Open' stand, respectively, for the newstest'14 and OpenSubtitlesTest test sets.

	PB-SMT				seq2seq				Transformer			
	PFSMB	MTNT	News	Open	PFSMB	MTNT	News	Open	PFSMB	MTNT	News	Open
Large 32K	22.7	22.1	16.1	27.4†	25.3	27.2	21.9	28.4†	27.8	28.5	27.1	31.9†

Table 4.2: BLEU scores for the Large training configuration using a 32K BPE vocabulary.

Surprisingly enough, it appears that the PB-SMT model outperforms both NMT architectures when translating PFSMB, the noisiest test set (please refer to Table 3.10 in Chapter 3). We can observe mixed results for the less-noisy UGC MTNT corpus. On the other hand, a consistent trend emerges: NMT consistently outperforms PB-SMT when translating canonical out- and in-domain test sets (newstest'14 and OpenSubTest). Overall, these results support the observations that NMT produces, *a priori*, the best results under in-domain evaluation conditions. However, such performance hides poor robustness for UGC test sets. This manifests itself as a performance gap between PB-SMT and NMT that reaches the highest value when comparing in-domain translation performance (+4.4 BLEU score on average[2]). In the same train of thought, a considerable performance decrease can be observed when evaluating another less related canonical corpus (+2.6 BLEU on average for newstest'14), and continues to decrease for the relatively clean MTNT UGC corpus (+2 BLEU) to finally be favorable to PB-SMT for PFSMB

[2]To compute the average performance gap, we consider the WMT and Small trained systems, defining newstest'14 and OpenSubTest and in-domain and OOD canonical test sets accordingly

(-3.1 BLEU).

These results are consistent with observations concerning the comparison of PB-SMT and NMT under low-resource conditions (Dowling et al. 2018), out-domain evaluation (Koehn and Knowles 2017) and noise robustness (Khayrallah and Koehn 2018); which highlight the relatively low performance of vanilla NMT systems compared to PB-SMT under these conditions. In the following sections, we will analyze these results to further highlight the difficulty of NMT models to generalize to UGC sentences by assessing their robustness capabilities.

Impact of vocabulary size We have also trained a version of each of our architectures, i.e. `PB-SMT`, `seq2seq` and `Transformer`, using a 32K vocabulary size in the LARGE OPENSUBTITLES training corpus to evaluate the impact of the BPE vocabulary size on translation quality. This was motivated by the observation that, as can be seen in Table 4.1, our NMT models performed considerably worse than `PB-SMT` with a 16K vocabulary on LARGE OPENSUBTITLES, even suffering a performance drop compared to SMALL OPENSUBTITLES, which shares the same domain and origin, while being substantially smaller than the former.

The results achieved by this system are reported in Table 4.2, for the larger vocabulary version (32K), PB-SMT seems to severely lose generalization capabilities: it achieves a similar performance for in-domain test sets, whereas performance on other test sets drops sharply. In contrast, NMT witnesses a generalization improvement. A similar observation has already been reported by Al-Haj and Lavie (2012) for PB-SMT, and is explained by an excessive token fragmentation, which limits the contextual information captured by the phrases.

In all experiments described in this work, we chose to keep the 16K BPE segmentation, as this setting results in the best performance for both of our training configurations.

4.3.3 Error analysis: are PB-SMT systems better than NMT architectures when processing UGC?

The goal of this section is to analyze both quantitatively and qualitatively the output of NMT systems to explain their poor performance when translating UGC. Several works have already identified two main limits of NMT systems: translation dropping (Sato et al. 2016), manifested as the production of sensible shorter prediction than the ground-truth translation, and excessive token generation, also known as over-generation (Roturier and Bensadoun 2011; Kaljahi et al.

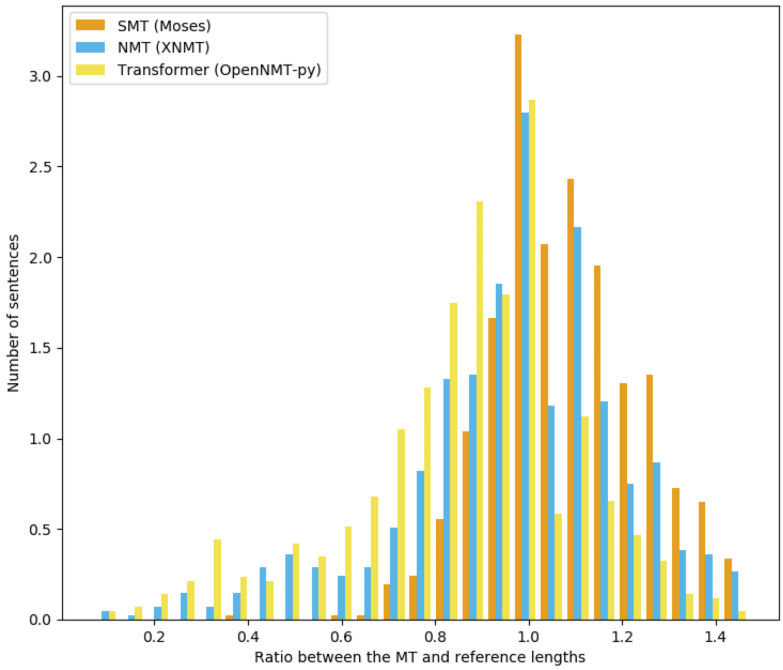


Figure 4.1: Distribution of `PFSMB` translations length ratio with respect to ground truth translations.

2015; Kaljahi and Samad 2015; Michel and Neubig 2018), in which the NMT systems output longer translation hypothesis than expected, typically producing repetitive sequences of tokens. This phenomena were also observed in more recent works (Ramesh et al. 2020), where NMT was found to produce translated token repetitions, omissions and spurious words when considering other low-resource scenarios. Next, we will analyze in detail how these two problems impact the MT models.

Translation dropping By manually inspecting the system outputs, we found that NMT models tend to produce shorter outputs than the translation hypotheses of the `PB-SMT` system, often avoiding translating the noisiest parts of the source sentence. For instance, the French sentence "`Bon je veux regard te _en w _olf [...]`"[3] *(Well I want to watch Teen Wolf [...])* is translated to a considerably shorter output by `seq2seq`, "`I want to look at you`", which results in an obvious change in meaning by the NMT system, also known as hallucinations (Raunak et al. 2021).

Analyzing the attention matrix for this observation, in Figure 4.2, shows that this problem is caused by a rare BPE token ("`te`"), part of the Named Entity "`teen wolf`", which is confused

[3] "_" indicates BPE pieces

①	src	**Vos gueul ac vos** Zlatan
	ref	**Shut the fck up with your** Zlatan.
	PB-SMT	**Your scream in your** Zlatan
	`seq2seq`	**Your shrouds with your** Zlatan
	Transformer	Zlatan!
②	src	**CE SOIR Y A L'ÉPISODE DE** #TeenWolf OMFGGGG
	ref	**TONIGHT THERE'S THE** #TeenWolf **EPISODE** OMFGGGGG
	PB-SMT	**Tonight's It At The EPISODE OF** #Teen Wolf OMFGGGG
	`seq2seq`	Teenwolf OMFGGGGGGGGGG
	Transformer	**THIS SOIRY HAS THE** #TeenWOL OMFGGGGGGGGGG
③	src	arretez de parler de teen wolf svp **:((**
	ref	stop talking about teen wolf please :(((
	PB-SMT	stop talking about teen wolf please **:((♪♪♪♪♪♪♪♪♪♪♪♪♪♪♪♪♪♪♪♪♪♪♪♪♪♪♪♪♪♪♪♪♪♪♪♪♪♪♪**
	`seq2seq`	Stop Tenwolf, please. **Please, please, please, please, please, please, please, yes, yes, yes, yes, yeah,**
	Transformer	Please stop!

Table 4.3: Examples from our noisy UGC corpus translated by our MT systems, displaying source (*src*) and reference translation (*ref*).

with the very common French pronoun "`te`" (*you* in English), where the attention weights are maximal across the sequence. As a consequence, the `seq2seq` model suddenly stops translating because the hypothesis "`I want to look at you`" is a very common English sentence with a much lower perplexity than the (correct) UGC translation. A similar pattern can be observed with the Transformer architecture in the case of rare token sequences on the source side, such as in examples ③ and ⑤ in Table 4.3, causing the translation to stop abruptly, whereas this is not the case for `seq2seq` and `PB-SMT`.

The phrase-based model, `PB-SMT`, does not suffer from this issue, as there is no entry in the phrase table that matches the sequence of BPE tokens of the source sentence. This illustrates how alignment tables can be more efficient than soft-alignment produced by attention mechanisms for highly noisy cases, in particular when the BPE tokenization generates ambiguous tokens, which confuses the NMT model.

In order to quantify the translation dropping phenomenon, we show, in Figure 4.1, the distribution of the ratio between the reference (ground-truth) translation sentence lengths and

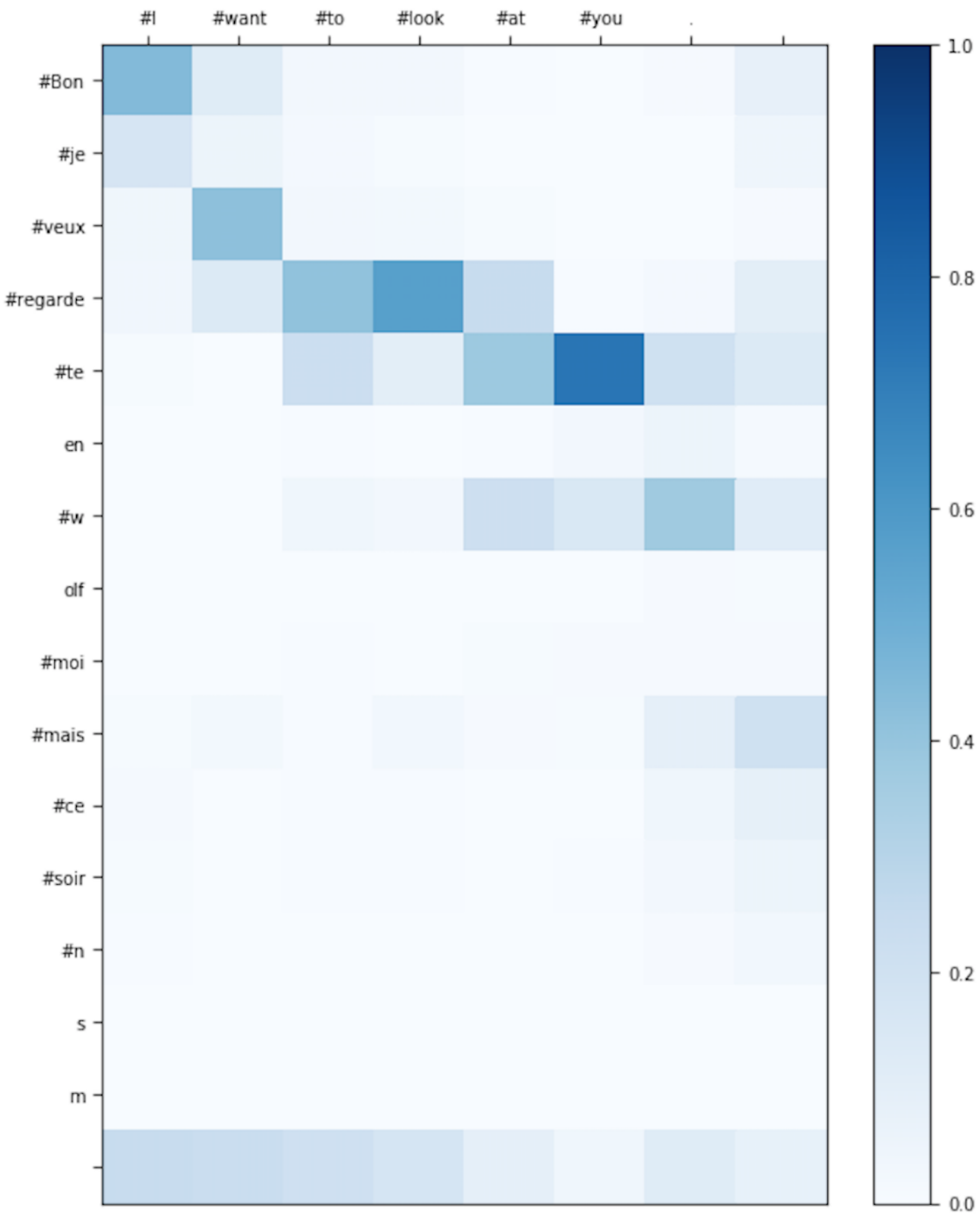


Figure 4.2: Attention matrix for the source sentence “Bon je veux regardé teen wolf moi mais ce soir nsm” *(Ok, I do want to watch Teen Wolf tonight motherf..r)* predicted by a `seq2seq` model.

the ones produced by PB-SMT and NMT for `PFSMB`. This figure shows that both NMT systems, (i.e. RNN-based and Transformers models), have a consistent tendency to produce shorter sentences than expected, while PB-SMT does not. These results show that off-the-shelf NMT systems produce overall shorter translations, as other authors have noticed (Zhang et al. 2020). Moreover, there is a substantial percentage of NMT predictions that are 60% shorter than the references, demonstrating the presence of translations that are dropped or shortened.

Over-generation A second well-known issue with NMT is that the model sometimes repeatedly outputs tokens that lack any coherence, thus adding considerable artificial noise to the output (Tu et al. 2016).

When manually inspecting the output, it can be noticed that this phenomenon occurred in UGC sentences that contain a rare, and often repetitive sequence of tokens, such as "`ne spooooooooilez pas teen wolf non non non et non je dis non`" *(don't spooooooil Teen wolf no and no I say no)*, in which the speaker's emotion is expressed by repeated characters (graphemic and punctuation stretching) and word repetition. These correspond to the "Marks of Expresiveness" UGC specificities category, reviewed in Section 3.3.

The attention matrix obtained when translating such sentences with a `seq2seq` model often shows that the attention mechanism becomes stalled due to the repetition of some BPE token, as shown, for example, in the attention matrix in Figure 4.3. More generally, it is noticeable that there are many cases in which the attention weights start increasingly focusing on the end-of-sentence token until the translation is terminated, while ignoring the source sentence tokens thereafter.

The transformer model exhibits similar problems (for instance, it translates the previous example to "`No no no no no no no no no no no no no no no no no no no`"). The `PB-SMT` system does not suffer from this problem and arguably produces the best translation: "`don't spooooooozt Teen Wolf, no, no, no, no, I say no`".

4.3.4 Source-side artificially amplified noise

As we just observed in the qualitative analysis and translation examples, noisy occurrences can affect the MT system's performance to a greater extent than the original source-side, e.g. single noisy tokens in the input can produce a larger-span negative impact on the system

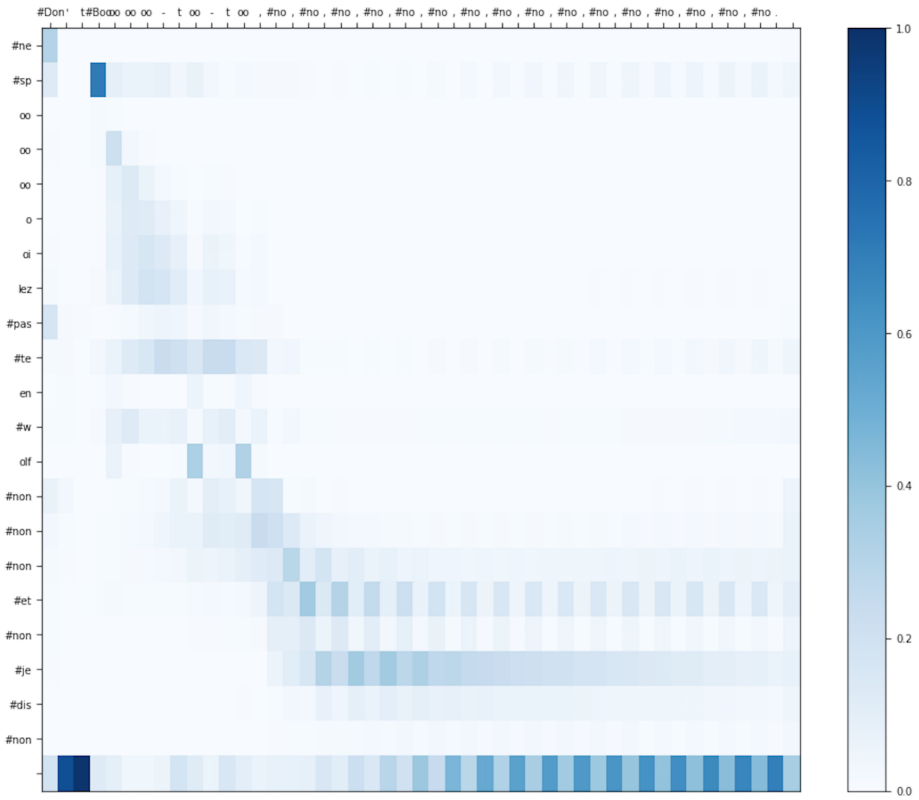


Figure 4.3: Attention matrix of a `seq2seq` model that exhibits the excessive token repetition problem. The sharp symbol (#) indicates spaces between words before the BPE tokenization.

representations of other tokens in the sentence. Therefore, to assess and quantify such a phenomenon and compare our MT systems within this scope, we report in Table 4.4 the Target-Source Noise Ratio (TSNR), introduced by (Anastasopoulos 2019) and previously described in Chapter 2.

As a reminder, a TSNR value greater than 1 indicates that the MT system adds more noise on top of the source-side noise, i.e. the rare and noisy tokens present in the source create even more noise on the output. Consequently, the higher the score, the more amplified the input-side UGC noise is (noisy artefacts caused by the MT system). That is, most of the noise and errors present in the translation hypotheses are due to errors of the MT system when processing noisy inputs.

This metric assumes that a corrected version of each source sentence is available. Hence, in order to compute this metric, we first manually corrected 200 source sentences of the `PFSMB` test corpus. By doing so, we produced a normalized canonical version of the French UGC source sentences, which is meant to be compared to the original (noisy) version. In Table 4.4, we

	PB-SMT	seq2seq	Transformer
WMT	4.62	5.00	4.92
Small	4.11	4.27	4.19
Large	3.99	4.27	4.09

Table 4.4: Noise added by the MT system estimated by TSNR for the PFSMB corpus, the lower, the better. Small and Large refer to the small and large instances of the OpenSubtitles training set.

report TSNR scores for our MT systems trained using our three training corpora, i.e. WMT, SMALL OPENSUBTITLES and LARGE OPENSUBTITLES. First, it can be noticed that for all systems, the value is considerably larger than 1 (between 4 and 5), which means that these MT models amplify the UGC noise occurrences in a proportion of roughly 4-fold, on average. Next, we can observe that PB-SMT has a better TSNR score, thus adding fewer artefacts (including dropped tokens and over-translations) to the output. It is also worth noting that the gap between the PB-SMT and NMT architectures (about 0.3 BLEU points) is much larger when training on WMT than when training in our OpenSubtitles (about 0.1 BLEU points). This could be explained by a more related linguistic register of the latter to social media discussion UGC, than that of the former. Indeed, the perplexity of the PFSMB corpus on a WMT language model is 826, compared to roughly 599 when using OpenSubtitles.

4.4 Conclusions and perspectives

In this chapter, we evaluate the capacity of different off-the-shelf MT architectures to translate UGC, and we perform a qualitative analysis of translations to understand when and why these systems lack robustness to UGC. An in-depth analysis of attention-based mechanisms depicts common causes of low translation performance and how UGC can “crash” the translation process.

Our experiments show that PB-SMT systems are more robust to noise than NMT models, and we provide several explanations about this relatively surprising fact. Concretely, we observed and investigated the discrepancy between BPE tokens as interpreted by the translation model at decoding time; and the addition of lexical noise factors that can get amplified during translation. We have also shown, by producing a new dataset with more variability, that using more training data was not necessarily the solution to cope with the specificities of UGC. The aim of this work

is, of course, not to discourage the NMT system deployment for UGC, but to better understand what aspect of PB-SMT methods contributes to noise robustness.

We have made our best effort to rule out the vastly documented scenario where NMT is in disadvantage compared to PB-SMT due to low-resource availability (Koehn et al. 2007b; Dowling et al. 2018), mainly caused by not large enough training corpus, regardless of the domain or noise breach between training and test sets. In this regard, we chose a fairly bigger training corpora than the reported minimum word quantity, which is roughly 20M (Koehn and Knowles 2017).

Our findings are in line with the NMT issues described in the works discussed in Section 3.8, where rare vocabulary or unusual token sequences often trigger bad or truncated translations by NMT. The conclusion of (Anastasopoulos 2019) that found a correlation between the performance detriment of NMT and the noise in the source sentence is confirmed. Nevertheless, we also show that PB-SMT seems much more robust, and its performance on highly noisy UGC corpora is much more stable performance-wise compared to both studied NMT architectures (RNN and Transformer).

Now that we have our MT baselines and first exploratory results of translating UGC, in the next chapter, we start searching for a simple and modular normalization baseline by conceiving a pre-processing pipeline that bridges the gap between UGC and canonical texts through correcting phonetic writing, to subsequently let the MT systems translate the normalized form. As we will see, this normalization pipeline leverages on the phonemic similarity between original UGC tokens and possible normalization candidates, which allows us to make corrections without any UGC normalization database.

Chapter 5

Addressing a frequent feature of French UGC: phonetic writing

Having our MT baselines and first results on translating UGC in the previous chapter, we propose a simple and modular normalization baseline. We design a pre-processing pipeline that bridges the gap between UGC and canonical texts through correcting noisy occurrences, and later having the MT systems translate the normalized form.

As discussed in Chapter 3, there is an extensive presence of phonetic writing in UGC (Moens et al. 2014), and the contribution of such a phenomenon has been studied specifically in French digital media writing styles (Wachs 2017). In this regard, we have developed an automatic normalization pipeline that leverages on the phonetic similarity between original UGC tokens and potential normalization candidates, which enables us to make corrections without any UGC normalization database. By doing so, we intend to recover from noise in the source by correcting tokens and, then, translate the normalized sentences using off-the-shelf MT systems.

In further detail, in this section, we present our first attempts to improve MT translation by proposing a re-ranking pipeline that aims to correct phonetic writing. As seen in Chapter 5, phonetic writing ultimately results in an increase of OOVs, e.g “j’ai fait” (I have done) $\rightarrow$ “g fé”, both phrases being homophones in French, although the latter is incorrect and does not exist in canonical texts. The phonetic writing UGC specificity can also cause incorrect syntactic use of known words, e.g. “j’ai regardé” (I watched) $\rightarrow$ “j’ai regarder” (I have watch). In this sense, this method aims to transform these unusual and non-canonical constructs into their canonical form,

which can be translated by any standard MT system in a subsequent stage.

Our contributions are threefold:

- We propose a pre-processing pipeline to normalize UGC and improve MT quality without using UGC training data in any form;
- by quantifying the corrections made by our normalizer in our UGC corpora, we quantify the presence of noise due to phonetic writing and demonstrate that using the phonetic information can be potentially exploited to produce corrections of UGC without the necessity of any kind of annotated data;
- we explore the performance improvement that can be achieved in machine translation by using a phonetic similarity heuristic to generate different normalization candidates.

The work presented in this chapter was presented at the W-NUT workshop (Núñez et al. 2019), co-located with EMNLP 2019.

5.1 Phonetic correction model

To automatically process phonetic writing and map UGC to their correct spelling, we propose a simple model based on finding, for each token of the sentence, words with similar pronunciations. Next, we select the best spelling alternative, using a language model. More precisely, our approach is made up of 4 steps:

1. Phonetizing: the pronunciation of each source token is automatically produced. All words in the input sentence as misspelled tokens are not necessarily OOVs (e.g. "*j'ai* ***manger***" — literally "*I have* ***eat***" — which must be corrected to "*j'ai* ***mangé***" — "*I have* ***eaten***", the French words "manger" and "mangé" having both the same pronunciation /mɑ̃.ge/);
2. Retrieving in-vocabulary tokens with similar pronunciation: using these phonetic representations, the method looks, for each word w of the input sentence, for every word in the training vocabulary with a "similar" pronunciation to w according to an ad-hoc metric we discuss below;
3. Encoding in Directed Acyclic Graphs (lattices): we represent each input sentence by a lattice of $n+1$ nodes, where n is the number of words in the sentence, in which the edge between the i-th and $(i+1)$-th nodes is labelled with the i-th word of the sentence.

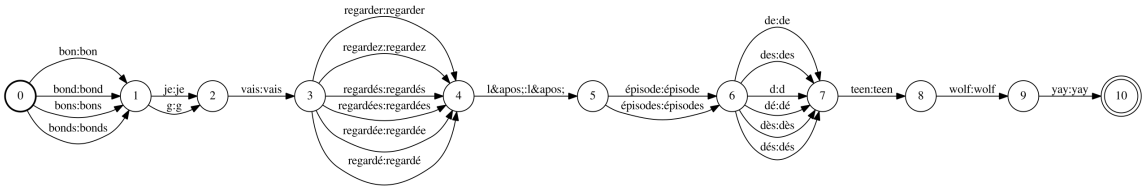


Figure 5.1: Example of lattice for a segment of a PFSMB UGC sample.

Alternative spellings can then be encoded by adding an edge between the i-th and $(i+1)$-th nodes, labelled by a possible correction of the i-th word. Figure 5.1 displays an example of such a lattice, showing the normalization candidates. In these lattices, a path between the initial and final nodes represent a (possible) normalization of the input sentence.

4. Normalization: using a language model, the probability to observe each alternative spelling of every normalized token of the sentence is computed (note that, by construction, the original UGC sentence is also contained in the lattice) and find the most probable path (and therefore potential normalization) of the input sentence. Note that finding the most probable path in a lattice can be done with complexity proportional to the size of the sentence, even if the lattice encodes a number of paths that grows exponentially with the sentence size (Mohri 2002). For these experiments, we used the OpenGRM (Roark et al. 2012) and OpenFST (Allauzen et al. 2007) frameworks that provide a very efficient implementation to score a lattice with a language model.

This process can be seen as a naive spell-checker, in which we only consider a reduced set of variations, tailored to the specificities of UGC texts. We will now detail the first two steps discussed above.

Generating the pronunciation of input words To predict the pronunciation of an input word, i.e. its representation in the International Phonetic Alphabet (IPA), we use the `gtp-seq2seq` python library[1] to implement a grapheme-to-phoneme conversion tool based on a `Transformer` model (Vaswani et al. 2017). We use a 3 layers model with 256 hidden units that is trained on a pronunciation dictionary automatically extracted from `Wiktionary` (our dataset is further described below). This vanilla model achieves a word-level accuracy of 94.6%, that is, it is able to find the exact correct phonetization of almost 95% of the words of our held-out data.

[1] https://github.com/cmusphinx/g2p-seq2seq

We also consider, as a baseline, the pronunciation generated by the `Espeak` program.[2] that uses a synthesis method to produce phonetizations based on acoustic parameters.

Finding words with similar pronunciation In order to generate alternative spellings for each input word, we look in our pronunciation dictionary for alternative candidates based on phonetic similarity. We define the phonetic similarity of two words as the edit distance between their IPA representations, all edit operations being weighted depending on the articulatory features of the sounds involved. Thus, to compute the phonetic similarity, we used the implementation (and weights) provided by the `PanPhon` library (Mortensen et al. 2016).

To account for peculiarities of French orthography, we also systematically consider alternative spellings in which diacritics (acute, grave, and circumflex accents) for the letter *"e"* (which is the only one that changes the pronunciation for different accentuation in French) were added whenever possible. Indeed, users often tend to 'forget' diacritics when writing online, and this kind of spelling error results in phonetic distances that can be large (e.g. the pronunciation of *bebe* and *bébé* is very different). We do not add diacritics on other French vowels, as accents do not change the French pronunciation for these, and, doing so increases the phonetic search space substantially.

We ultimately only keep as candidates words those that are present in the training corpus presented in Chapter 4 to filter out OOV and non-existent words.

Pronunciation dictionary To train our Grapheme-to-Phoneme model, we use a dictionary mapping words to their pronunciation (given by their IPA representation). To the best of our knowledge, there are no easily accessible pronunciation dictionaries for French. In our experiments, we have considered a pronunciation dictionary automatically extracted from `Wiktionary` dumps based on the fact that, at least for French, pronunciation information is identified using special *templates*, making its extraction straightforward (Pécheux et al. 2016).

The dictionary extracted from French `Wiktionary` contains 1,571,090 words. We trained our `G2P` phonetizer on 1,200,000 examples, leaving the rest to evaluate its performance. When looking for words with similar pronunciation, as discussed in the previous section, we consider only the words that appear in our parallel MT training data (described in Chapter 4) to speed

[2] `espeak.sourceforge.net`

up the search and rule out any possible OOV during evaluation. After filtering, our dictionary contained pronunciation information for roughly 82K French words.

5.2 Statistics of the normalization process

Table 5.1 reports the most common normalization changes performed by this method in the PFSMB test set (described in Chapter 3). It can be noticed that they are mainly grammatical corrections of badly used in-vocabulary tokens with identical French pronunciation.

Normalization	a→à	sa→ça	et→est	la→là	à→a	tous→tout	des→de	regarder→regardé	ils→il	prend→prends
Number of app.	87	16	15	13	12	11	8	7	6	6

Table 5.1: Most frequent normalization replacements on the PFSMB test corpus.

On the other hand, in Figure 5.2, we show the effect of changing the maximum phonetic distance threshold in order to consider that any given word in the training word-level vocabulary should be retained as a normalization candidate in the lattice. It is straightforward to see that a higher value of this parameter leads to a linear increasing quantity of word-level replacements. However, due to the combinatorial nature of the lattices, a value that is too high has, as a direct outcome, an exploding quantity of candidate normalization phrases, and consequently, a higher computational cost.

To avoid an explosion of the number of alternatives that we consider, and upon experimentation, we determined that the best phonetic distance threshold value was 0.1 for these experimental conditions, which, as can be seen in Figure 5.2, corresponds to keeping only homophone normalization candidates, in which we could observe the most conservative normalization without introducing excessive noise to possible DAG paths, as discussed previously. This results, as can be seen in the figure, in a total of 600 replacements that take place in the PFSMB test set, which corresponds to changing 5% of the original tokens.

5.3 Machine translation results

Table 5.2 show the MT BLEU performances, ultimately produced by the proposed method along with the baselines, PB-SMT and Transformer MT systems, described in Chapter 4, and whose results are reported again to make comparison easier.

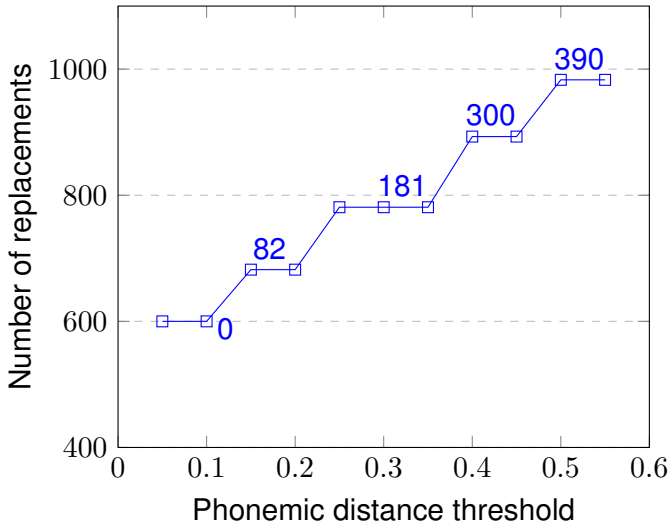


Figure 5.2: Number of replacement operations of our normalizer over the PFSMB test set. The quantity of non-homophones normalizations are displayed as point labels.

We noticed a significant improvement in the results for the UGC test corpora when using the Transformer architecture trained with the SMALL OPENSUBTITLES training set. Specifically, a BLEU score improvement for the PFSMB and MTNT test corpora when using G2P and Espeak phonetic normalization in Table 5.2, compared to the baseline MT systems. Interestingly, these improvements only hold for the Transformer model, whereas we consistently obtain a slight decrease of BLEU scores when the normalized text is translated using the PB-SMT model. Concisely, we noticed an improvement of +1.5 BLEU for PFSMB using the G2P phonetizer, and +0.5 for MTNT when using the Espeak phonetizer instead.

		PB-SMT				Transformer			
		PFSMB	MTNT	News	Open	PFSMB	MTNT	News	Open
Baseline	WMT	20.5	21.2	**22.5**†	13.3	15.4	21.2	27.4†	16.3
	Small	28.9	27.3	20.4	26.1†	27.5	28.3	**26.7**	31.4†
	Large	**30.0**	**28.6**	22.3	**27.4**†	26.9	28.3	26.6	**31.5**†
G2P	WMT	20.4	20.2	**21.9**†	13.4	15.0	20.4	**26.7**†	16.2
	Small	28.4	26.2	19.9	26.1†	**29.0**	28.3	25.7	31.4†
	Large	**29.0**	27.6	21.8	**27.4**†	28.5	28.2	25.9	**31.5**†
Espeak	WMT	20.4	20.4	21.7†	13.4	14.6	20.7	26.5†	16.1
	Small	28.0	26.3	19.8	26.2†	28.5	**28.8**	25.6	31.4†
	Large	28.3	**27.7**	21.6	**27.4**†	27.5	28.6	25.8	**31.5**†

Table 5.2: BLEU score results for our three benchmark models on baselines (without normalization) and normalized test sets using G2P and Espeak phonetizers. The best result for each test set is marked in bold, in-domain scores with a dag.

System	Blind Tests MTNT	PFSMB
Large - PB-SMT Raw	**29.3**	**30.5**
Large - PB-SMT Phon. Norm	26.7	26.9
Small - Transformer Raw	**25.0**	**19.0**
Small - Transformer Phon. Norm	24.5	18.3
M&N18 Raw	19.3	13.3
M&N18 UNK rep. Raw	21.9	15.4

Table 5.3: BLEU score results comparison on the MTNT and PFSMB blind test sets. The G2P phonetizer has been used for normalization. *M&N18 stands for (Michel and Neubig 2018)'s baseline system.*

Regarding the performance decrease in the canonical test corpora, newstest'14 and OpenSubtitles, we can observe that there is usually a considerable under-performance on the latter (-0.65 BLEU averaging over the 6 models and training set configurations), which is not as noticeable in the former (-0.1 BLEU in the worst case). This could be explained by the substantially longer sentences in newstest'14 compared to OpenSubtitles, which have roughly 6 times more words on average, according to Table 3.9 in Chapter 3. That is, when sentences are longer, the number of possible paths in the lattice grows exponentially, thus increasingly adding confusion to the language model's decisions, which will ultimately produce the most probable normalization. This observation strongly suggests that the performance of the normalizing method depends on the length of the target sentence that is to be normalized.

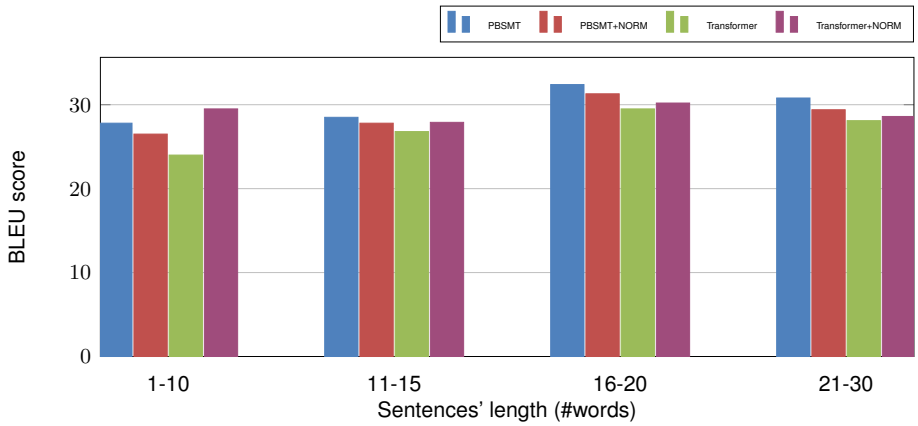


Figure 5.3: Bar plot of the BLEU score for the PFSMB test set. The translation hypotheses are divided into sentences' length groups.

Motivated by this observation, we have also calculated the BLEU score of the PFSMB corpus by groups of sentences' length in order to further investigate why this method improves

the translation quality of Transformer but not for PB-SMT models. The results reported in Figure 5.3 show that the highest improvement resulting from phonetic normalization is present in short sentences (between 1 and 10 words). It is worth noting that this is the only situation in which Transformer outperforms PB-SMT. Hence, the higher overall Transformer BLEU score over PB-SMT is certainly due to a relatively high successful normalization over the shortest sentences of the PFSMB test set. This is in line with the documented fact that NMT is consistently better than PB-SMT on short sentences (Bentivogli et al. 2016) and, in this concrete example, it seems that Transformer can take advantage of this when this normalization pipeline is applied. Furthermore, these results could be considered as evidence supporting the conclusion that the proposed method generally performs better for short sentences, as observed in Table 5.2.

Furthermore, we have applied our method to a set of blind tests of UGC corpora MTNT and PFSMB. These results are displayed in Table 5.3; we also show the performance of the (Michel and Neubig 2018)'s baseline system on such test sets. The translation system is selected as the best for each of the UGC sets in Table 5.2. For this test corpus, we noticed a 0.5 and a 3 BLEU point decrease for the Transformer and PB-SMT systems, respectively, when our normalizer is used on the MTNT blind test. On the other hand, we obtained a 0.7 BLEU point loss for Transformer and a 3.6 point drop for PB-SMT, both evaluated on the PFSMB blind test. These results suggest that, when we do not tune our method (using an adequate UGC development set) looking for the best translation system, and for certain UGC sets, our approach introduces too much artificial noise, and MT performance can therefore be negatively impacted.

5.4 Qualitative analysis

Table 5.4 reports some examples of the output of this method, along with their translation before and after correction by our phonetic normalization process.

For Example ① in Table 5.4, it can be observed that the normalizer enables the MT system to produce the first part of the translation ("*When I get to the taff*") correctly. This is the result of correcting the French homophones "*arriver*" into "*arrivé*", i.e. from the infinitive to the past form. It is very interesting to note that the robustness of the Transformer using subword units, seems to be good enough to correctly translate the typographical error "*ce met a battre*", thus, the correct proposed normalization ("*se met à battre*") does not impact the MT result but

①	src	**arriver au taff, des que j'ouvre le magasin** je commence **a** avoir le vertige mon coeur **ce met a battre a 200** et je sens que je vais faire un malaise,
	norm src	**arrivé au taff, dès que j'ouvre le magasin** je commence **à** avoir le vertige mon coeur **se met à battre à 200** et je sens que je vais faire un malaise,
	ref	**once at work, as soon as I open the store** I'm starting to feel dizzy my heart **starts racing at 200** and I feel I'm gonna faint,
	raw MT	I start to get dizzy. My heart starts to beat at 200 and I feel like I'm going to faint.
	norm MT	**When I get to the taff, as soon as I open the store**, I start to get dizzy. My heart starts pounding at 200 and I feel like I'm gonna get dizzy.
②	src	c un peu plus que mon ami qui **faite son annif**,
	norm src	c'est un peu plus que mon ami**e** qui **fête son annif**
	ref	it's a bit more than a friend to me who **celebrate his birthday**,
	raw MT	It's a little more than my friend **doing his birthday**,
	norm MT	It's a little more than my friend **celebrating her birthday**,
③	src	zlatan est **nez** pour **marqué**
	norm src	zlatan est **né** pour **marquer**
	ref	Zlatan was **born** to **score**
	raw MT	Zlatan's **nose** is for **marking**
	norm MT	Zlatan was **born** to **score**
④	src	**Cartes bancaires** de Zlatan retrouvés dans un taxi... On en parle ou pas WWW44
	norm src	**Carte bancaire** de Zlatan retrouvé dans un taxi... On en parle ou pas WWW44
	ref	Zlatan's **bank cards** found in a cab... we talk about it or not WW44
	raw MT	Zlatan's **bank cards** found in a cab... we talk about it or not WW44
	norm MT	Zlatan **bank card** found in a taxi... we talk about it or not WWW44

Table 5.4: Examples from our noisy UGC corpus normalizations. *We show the original UGC source (src), reference translation (ref), the normalized source produced by our approach (norm src), the translation produced by* `Transformer` *from the original source (raw MT) and the one using the normalized source (norm MT).*

certainly has an effect over the correctness of the French phrase.

Regarding Example ② in Table 5.4, it should be noted that the normalized proposition significantly improves MT translation, producing an output closer to the reference translation, compared to the raw MT output. The key normalization change is the misused French token "*faite*" (pronounced /fɛt/) — "*does*" in English — by its correct homophone "*fête*" — "*celebrates*" in English. It can also be noticed that the robustness of the MT system is once again capable of correctly translating a phonetic writing contraction "*c*" as the two correct tokens "*c'est*".

Example ③ in Table 5.4 shows how semantically different can be a misused French word due to confusing homophones. We can observe that the normalization replacement *"nez"* ("nose" in English) → *"né"* ("born" in English), which are French homophones, drastically changes the meaning of the output translation. Additionally, the correction *"marqué"* → *"marquer"*[3] (changing to correct verb tense) also causes the translation to be closer to the reference.

Finally, in Example ④ in Table 5.4 we show some caveats and limitations for our proposed method, where the correct original plural "*Carte**s** bancaire**s** ... retrouvé**s***" was changed to the

[3] *marked* vs. *mark*-INFINITIVE in English.

singular form "*Carte bancaire ... retrouvé*". This is due to the homophonic pronunciation of most French singular and plural pronunciations. Whenever there is no discriminant token with different pronunciation, such as an irregular verb, the language model has trouble choosing the correct final normalized phrase since both plural and singular propositions are proposed as candidates. Thus, either can be indistinctly kept as final normalization, since both forms are correct and theoretically very similar in their perplexity measure.

5.5 Conclusions

In this chapter, we have proposed a pre-processing method that relies on phonetic similarity to normalize UGC. Our method can improve the translation quality of UGC of modern off-the-shelf NMT systems. Conversely, we have performed an error analysis showing that the MT system successfully translates phonetic-related errors with its increased robustness. However, it must be noted that we obtained negative results on a blind test evaluation, suggesting that the phonetic normalization approach introduced more noise than useful corrections on totally unseen data. This highlights the importance of holding out data in the form of blind test sets so that the real efficiency of an MT system can be verified. In addition, we have applied our normalizer to clean canonical test data and have shown that it slightly hurts MT performance. More in-depth studies are needed to assess whether our proposed normalization pipeline can correct phonetic-related errors in UGC for other languages and other difficult UGC scenarios, such as video game chat logs (Martínez Alonso et al. 2016), while maintaining good performance on canonical texts.

Although our normalization pipeline showed some interesting corrections and helped NMT recover from errors that otherwise it could not, it also turned out to add artificial noise to some extent and displayed poor generalization over our blind test, showing a negative impact for the `PB-SMT` translation system. In this sense, our proof-of-work could benefit from other heuristics and constraints for the re-ranking, as well as further investigating and refining the phonetic distance measuring.

Subsequently, we decided to focus our efforts on exploring other alternatives to cope with UGC by using end-to-end learning representations of NMT models to account for a wide spectrum of different specificities (discussed and listed in Section 3.3) that can be resolved using morphological cues, instead of pinpointing a single type of UGC idiosyncrasy. We thus

investigate finer granularity-level translation (i.e. char-based) to assess how well char-based performs when translating UGC compared to the coarser translation granularities used so far in the dissertation. Furthermore, we also realized that a single corpus-level BLEU score was not helpful to assess whether the proposed MT models perform better or worse than the baselines with respect to any given UGC specificity. This led us to propose, in the next chapter, a UGC evaluation framework that links performance change with concrete UGC specificities, uncovering many of the performance and robustness caveats and advantages of a given MT system in terms of well-defined UGC specificities.

Chapter 6

Character-level for noisy UGC MT

After exploring our first automatic normalization method tailored to one UGC specificity (phonetic writing), we decided to study and improve the generalization properties of NMT systems by leveraging fine-grained translation granularity. This approach, compared to the one introduced in the previous chapter, aims to address a wider range of UGC specificities that the MT system could correctly translate by relying on models that operate at the character level. Indeed, character-based models are open-vocabulary models designed specifically to learn n-grams representations (and therefore translations) of tokens. Segmentation at the character-level is one of the standard approaches to deal with the OOV problem in NLP systems as it is, in contrast to handling OOVs using an a unique `<UNK>` token (discussed in Section 2.3), capable of producing a learning representation for any sequence of seen characters.

In order to investigate the impact of these models on UGC translation, we compare the robustness of characters-based systems and that of BPE-based systems. We do not consider coarser granularities, such as words, since it has been proven that BPE tokenization consistently outperforms word-level segmentation applied to MT, as reviewed in Section 2.3. Our intuition is that, by conceiving robust-enough NMT systems that take advantage of character-level compositions, the architecture could associate neural representations of noisy OOVs (e.g. spelling errors) to 'clean' tokens, therefore automatically recovering from noise.

In a second step, building on the results of our first experiments, in Section 6.3 we show, that despite their word-level open-vocabulary properties, the presence of out-of-vocabulary characters (`charOOVs`), a characteristic of UGC (refer to Chapter 3), hurts translation quality of character-based models and questions our intuition about these models. In order to elaborate

on this issue, we study the open-vocabulary capacities of character-based models and proposed a simple way to deal with `charOOVs`, by strongly shortening the character vocabulary size, which we also found had low impact on canonical and in-domain translation. Specifically, our approach uses substantially smaller character vocabulary size than standard character-level model configurations constitutes a call for action to correctly tune the vocabulary size, a parameter that has been overlooked in the literature.

During our experiments, we noticed that, in order to correctly and better assess the advantages of our proposed methods for translating UGC, a single metric that evaluates the translation for a corpus with several confounded phenomena was not sufficient to identify how distinct UGC specificities are handled differently by a given model. This led us to develop a UGC evaluation framework that aims to rigorously characterize the capabilities of an MT system to cope with UGC evaluation. This framework is described in Section 6.5.1.

In the final part of this chapter (Section 6.7), we combine character-level NMT with a phoneticized version of the source to produce the target translation characters, which we refer to as `phon2char`, and whose robustness capabilities are also assessed using our evaluation framework.

6.1 Why character-based MT?

We intend to use character-level NMT approaches to investigate to what extent they are robust to UGC specificities and noise at test time. Character-based MT seems to be a promising method to translate UGC because of its open-vocabulary property: character-based systems can model the inner semantics of potential word-level OOVs and will translate them, instead of simply generating a `<UNK>` token as in usual word or sub-word tokenizations.

This feature is appealing for translation when spelling errors and, more generally, OOVs, can be mapped to canonical (sequences of) tokens that are present during training. Indeed, in this case, the usual methods for treating OOV (copying them from the source using alignment-based methods and the `<UNK>` token) will not attempt to translate them. It should however be noted that implementing this intuition has to avoid a potential pitfall: in UGC many OOV are named entities (such as hashtag, people names, etc.) that are often the same in the source and target languages.

6.2 Character-based models

As discussed in Section 2.3, character-level models pose inherent challenges, namely, increased computational costs and long-range dependency modeling. In order to overcome these difficulties, convolutional neural network encoding has been explored to get the best of both worlds, i.e. maximizing the capacity-to-compression ratio (Cherry et al. 2018) of a given NMT base architecture. We investigate two of these character-based architectures. The first model we consider, `charCNN` (Kim et al. 2016), is a classic encoder-decoder in which the encoder uses character-based embeddings in combination with convolutional and highway layers to replace the standard lookup-based word representations. The model considers, as input, a stream of words (i.e. it assumes the input has been tokenized beforehand) and tries to learn a word representation that is more robust to noise by unveiling regularities at the character level. Thus, it should be noted that, despite learning character-based representation for the source words, `charCNN` processes the input and decoding at the word level. This architecture was initially proposed for language modeling Costa-jussà and Fonollosa (2016); shows how it can be used in an NMT system and reports improvements up to 3 BLEU points when translating from a morphologically-rich language, German, to English.

The second model we consider does not rely on an explicit segmentation into words: Lee et al. (2017) introduce the `char2char` model that directly maps a source character sequence to a target character sequence without any segmentation (spaces being considered as a vocabulary element) thanks to a character-level convolutional network with max-pooling at the encoder. It can be considered as an open-vocabulary model: it can generate any word made of any of the N most frequent characters of the train set (where N is a model hyperparameter) and only outputs a `<UNK>` token in the presence of a character that is not in this (char-) vocabulary. Lee et al. (2017) show that this model outperforms subword-level (i.e. BPE-based) translation models on two WMT'15 tasks (De-En and Cs-En) and gives comparable performance on two tasks (Fi-En and Ru-En). Lee et al. (2017) additionally report that in a multilingual setting, the character-level encoder significantly outperforms the subword-level encoder on all the language pairs.

	WMT				OpenSubtitles			
	PFSMB	MTNT	News†	OpenSubTest	PFSMB	MTNT	News	OpenSubTest†
BPE-based models								
seq2seq	9.9	21.8	27.5	14.7	17.1	27.2	19.6	28.2
+ <UNK> rep.	**17.1**	**24.0**	**29.1**	**16.4**	26.1	**28.5**	24.5	28.2
Transformer	15.4	21.2	27.4	**16.4**	**27.5**	28.3	**26.7**	**31.4**
Character-based models								
charCNN	6.2	12.7	17.2	9.2	13.3	16.3	10.1	21.7
+ <UNK> rep.	16.1	18.2	22.1	11.5	18.6	20.2	14.6	23.9
char2char	7.1	13.9	18.1	8.8	23.8	25.7	17.8	26.3

Table 6.1: BLEU score for our models for the different train-test combinations. In-domain test sets are marked with a dag. 'News' and 'Open' stand, respectively, for the WMT and OpenSubtitles test sets. WMT and OpenSubtitles are the training corpora, described in Section 3.5

6.3 Results

Table 6.1 reports the BLEU scores of the different character- and BPE-based models that we consider. As observed in Section 4.3, the simple replacement strategy for the <UNK> token substantially increases the BLEU score, which also benefits the word-level decoding of charCNN, whereas char2char, with a character-by-character output, and using the original implementation parameter, does not output any <UNK>.

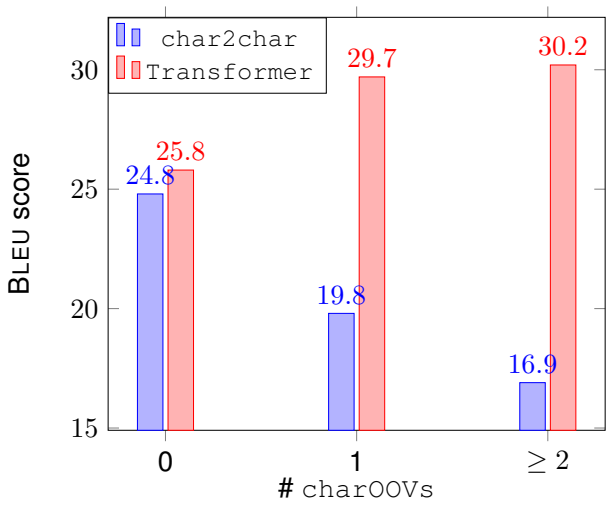


Figure 6.1: BLEU score in function of the number of charOOVs in the input sentence.

As expected, all models perform better when trained on the OpenSubtitles corpus than when trained on WMT, as the former is intuitively more similar to UGC data than the latter. Moreover, it appears that character-based models are largely outperformed by BPE-based models for most train-test combinations and, therefore, that their relative capacity to learn word

representations that are robust to noise can be questioned. We explore some ways to reduce the performance gap between these two NMT approaches in the following.

Another interesting observation is that, while Transformer achieves the best results in all test sets when trained using OpenSubtitles, it is outperformed by seq2seq on the WMT training configuration. The situation is similar for character-based models. This observation suggests that these models do not capture the same kind of information and do not have the same generalization capacities, as they roughly have the same number of parameters (69M parameters for the char2char and Transformer models, 65M for the charCNN and 49M for the seq2seq model).

Finally, the impact of charOOVs on BLEU performance is shown in Figure 6.1, comparing the results for the Transformer and char2char models for different numbers of such tokens in the inputs of the PFSMB test corpus. It can be seen that the former is considerably more sensitive to occurrences of charOOVs, showing a consistent performance detriment as there are more of them in the inputs. In turn, Transformer is increasingly good when more charOOVs appear in the sources, which can be explained by the fact that most charOOVs are left unchanged between the source and the reference (e.g. emojis, # and @). In this sense, Transformer shows a clear advantage due to its ability to manage <UNK> tokens, present during training to some extent, whereas the char2char system does not produce any <UNK>, being unable to output charOOVs, and, furthermore, these occurrences affect the rest of the input sequence, leading the model to produce wrong translations as shown in Table 6.3.

Error analysis In order to find which kind of UGC specificities are the most difficult to translate and can explain the difference in performance between character-based and BPE-based systems, we have conducted a contrastive analysis between the predictions of the Transformer and the char2char models. For each system, we have selected the 100 source sentences with the best translation and the 100 ones with the worst translation.[1] We have manually annotated these 400 sentences, using the typology described in Table 6.2, to identify which UGC specificities were the hardest to translate. Examples of annotations are given in Table 6.3.

For instance, the char2char model only outputs 8 sharp symbols when translating the test

[1]The translation quality was simply defined as the edit distance between the translation hypothesis and the reference translation.

set of the PFSMB, whereas the reference, contains 345 hash tags starting with a '#'.

While the Transformer model is less sensitive to this problem (it produces 105 sharp symbols when translating the PFSMB test), its translation quality also suffers from the presence of hashtags as it often tries to translate part of the hashtags relying on the sub-word units identified by the BPE tokenization rather than simply copying them from the source. Additionally, it can be noticed that errors 2 and 12 (diacritization and graphemic/punctuation stretching) are treated somewhat better by the char2char model than by the Transformer model, being less frequent in the worst translations of the former.

Figure 6.2 shows the number of UGC specificities in each of the 100 worst and best translations of the two considered models.[2] For both models, the most difficult specificities appear to be the presence of NE (category 10) and the inconsistent casing (category 8) often corresponding to several words written in full uppercase to denote emotions or excitement. Interestingly, these two types of noise have more impact on the char2char model than on the Transformer model, even if it could be expected that the character embeddings learned by the former would not be sensitive to the case. Another important category of errors is category 6 that corresponds to hashtags, mentions, and URLs, for which the char2char model is not capable of producing characters or a sequence of characters that are rare in the training set (namely, #, @ or http://www).

code	kind of specificities
(1)	Letter deletion/addition
(2)	Missing diacritics
(3)	Phonetic writing
(4)	Tokenization error
(5)	Wrong verb tense
(6)	Special chars. (#, @, URL)
(7)	Wrong gender/number
(8)	Inconsistent casing
(9)	Emoji
(10)	Named Entity
(11)	Contraction
(12)	Graphemic/punct. stretching
(13)	Interjection

Table 6.2: Typology of UGC specificities used in our manual annotation scheme. Refer to Section 3.3 for further details.

Finally, in Figure 6.1, it can be seen that for the PFSMB phrases that do not have any charOOVs, the char2char BLEU score is much similar than that of the Transformer in Table 6.1. Concretely, the performance gap is reduced from 3.7 BLEU points to 1.9 in the absence of charOOVs, while other UGC specificities are still present. This encourages us to further investigate the char2char behavior when charOOVs are present in the input, which

[2]Similar conclusions can be drawn from the observation of the 100 best translations (see Figure 1 in supplementary material for details).

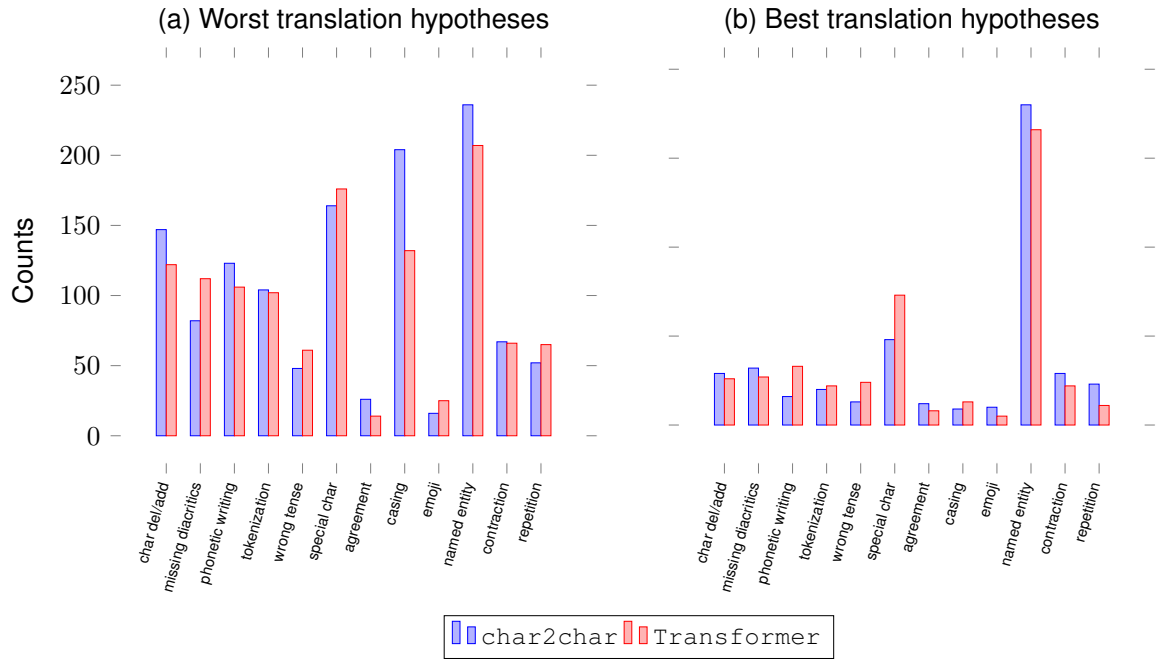


Figure 6.2: Comparison of the number of UGC specificities in the best and worst translation hypotheses of `char2char` and `Transformer`. Noise categories are defined in Table 6.2.

we further investigate using the copy task control experiment below.

Qualitative analysis Table 6.3 reports translation hypotheses predicted by the `Transformer` and `char2char` models. These examples illustrate the difficulty of the `char2char` model to translate named entities: while the simple strategy of copying unknown OOVs from the source implemented by the `Transformer` is very effective, the `char2char` model tends to scramble or add letters to NE. They also illustrate the impact of phonetic writing on translation: for instance "*rescend*" that should be spelled "*ressents*" (example 4) and "*joue a*" that should be spelled "*joue à*" (example 5) both result in a wrong translation: in the former case, the meaning of the sentence is not captured, and in the latter the "*a*" is wrongly copied in the translation hypothesis.

6.4 Copy task control experiment

To corroborate our analysis of the impact of special characters on `char2char` and quantify the impact of `charOOVs` and of rare character sequences, we conduct a control experiment in which a `char2char` system with different vocabulary sizes must learn to copy the source, that is to say: we train the `char2char` model on an artificial parallel corpus in which the source and

①	src	**JohnDoe389 (10)** qui n'arrive pas **a (2) depasser (2)** 1 a **(2) FlappyBird** ... **ptddddr (11, 12)**
	ref	JohnDoe389 who can't score more than 1 at FlappyBird ... lmaooooo
	Tx	**John Doe389** who can't pass one **to** Flappy Bird... **ptdddr.**
	c2c	**Johndeigh** doesn't happen to pass one **on** Flappyrib... **please.**
②	src	**JohnDoe665 (10) viens (5)** de regarder **Teen Wolf (2011) S03E17 [Silverfinger] (10)** et s'en va apprendre l'humanité à **Castiel (10)**.
	ref	JohnDoe665 just watched Teen Wolf (2011) S03E17 [Silverfinger] and he's on his way to teach Castiel humanity.
	Tx	**John Doe665 just watched** Teen Wolf (2011) S03E17 (Silverfinger) and is going to teach humanity at Castiel.
	c2c	**Johndoedsoids is looking** at Teen Wolf 2011, and learn about humanity in Castiel.
③	src	**Jai (1,4)** fait 10 **a (2)** flappy bird **(10) mddr (11, 12) # JeMeLaDonneMaisJavancePas (6)**
	ref	I did 10 at flappy bird lool # JeMeLaDonneMaisJavancePas
	Tx	**I did** 10 **a** flappy bird mdr **# I'm not moving**
	c2c	♪**I've made** 10 flappy birdd ♪
④	src	Si au bout de 5 mois **tu rescend (3) toujours se (3) genre** "d'effet secondaire" c'est vraiment mauvais.
	ref	If after 5 months you're still feeling this kind of "side effect" that's really bad.
	Tx	If after five months you're still **re-exciting this** whole "special effect" thing is really bad.
	c2c	In the end of five months, you're **always responding** to secondary effects, that's really bad.
⑤	src	y a ma cousine qui **joue a (2)** flappy bird **(10) mdrrrrrrrrrr (11, 12) elle et plus nuuul (12,7) que moi**
	ref	my cousin plays flappy bird looooooooooool she's more hopeless than me
	Tx	There's my cousin who **plays a** flappy bird **mdrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr**
	c2c	There's my cousin that **plays a** flappy bird **might bring her and nouse than me**.

Table 6.3: Examples from our noisy UGC corpus showing the `Transformer` (denoted as Tx) and `char2char` (denoted as c2c) predictions. Source sentences have been annotated with UGC specificities of Table 6.2 (in blue) according to their numerical code. Parts of the reference that were correctly translated are underlined and noise occurrences and their translations are marked in bold.

target sentences are identical. By varying the number of characters considered by the system, we can control the number of rare characters in our corpora (recall that with a char-vocabulary of size N all characters, but the N most frequent ones are mapped to a special <UNK> character). Note that this task is a simplification of the problem we have highlighted in the previous section: the model simply has to learn to always make an exact copy of the whole source sentence and not to detect some special characters (such as # or @) from the input that trigger the copy of the next characters, while the rest of the sentence should be translated.

More precisely, we built an artificial train set containing 1M random sentences with lengths between 5 and 15 characters long, keeping a 164 fixed-size character vocabulary (this corresponds to the size of the extended ASCII alphabet), and whose characters are distributed uniformly, in order to rule out the impact of rare characters and keeping only the effect of `char-OOVs` over the performance. We consider two test sets made of 3,000 sentences each: `in-test` that uses the same 164 characters as the train set and `out-test` that uses 705 different characters. Source and reference are identical for every example of the train and test sets.

Results Table 6.3 reports the results achieved on the copy task with and without replacing the predicted <UNK> symbols.

Note that, in this very simple task, <UNK> characters are always replaced by their true value. These results show that this task is not trivial for char-based systems: even when all characters have been observed during training, the system is not able to copy the input perfectly.

Above all, reducing the vocabulary size from 164 to 125 results in an increase of the BLEU score on the two considered conditions, even without replacing the <UNK> that the system has started to generate, where '%<UNK> pred.' indicates the percentage of <UNK> tokens in the prediction. Further reducing the size of the vocabulary artificially improves the quality of the systems: they generate more and more <UNK>, which are replaced during post-processing by their true value. These observations suggest that unknown or rare characters are not only difficult to copy, but they also distort the representation of the input sequence built by the system, impacting the whole prediction.

	Vocabulary Size				
	164	125	100	80	60
in-test					
%<UNK> pred.	0	0.2	5	17	29.5
BLEU	92.9	95.8	77.6	24.9	1.9
+<UNK> rep.	92.9	96.6	98.4	98.5	98.7
out-test					
%<UNK> pred.	0	9.2	13.8	25	36
BLEU	54.5	63.7	52.3	15.3	0.9
+<UNK> rep.	54.4	96.6	98.7	99.1	99.5

Figure 6.3: Results of the copy task, evaluated by the BLEU score before and after <UNK> replacement (*+<UNK> rep.*) and percentage of <UNK> characters in the prediction (*%<UNK> pred.*).

6.5 Robustness impact of UGC's specificities

In order to quantify and link robustness and performance impact of NMT when processing noisy UGC, we annotated 400 random-sampled PFSMB source sentences according to the typology shown in Table 6.2, including a software framework to control the type and occurrences of noise in the source, and, automatically producing a comparable reference for evaluation.

The PMUMT corpus To understand the impact of UGC peculiarities we created the Phenomena Modeling UGC Machine Translation (PMUMT) by manually annotate 400 source sentences sampled from the PFSMB: one of the authors, fluent in French and with good knowledge of UGC, has identified spans in the sentence that differ from canonical French and characterized these specificities using the fine-grained typology of Sanguinetti et al. (2020) (see Table 6.2). Since the whole annotation process was done by a single person, no inter-annotator agreement can be calculated. Nevertheless, results of our pilot study for each individual UGC peculiarity (cf. Table 6.5 for a cross-metrics analysis), show that MT performance consistently performs better on our normalized corpus than on the original noisy set.

Each span containing a UGC specificity has been 'normalized' to a form closer to canonical French.[3] Table 6.4 shows some examples of annotated (source) sentences. Additionally, a normalized form of each target (i.e. English) sentence has also been produced to ensure that

[3]To ensure that this normalization has actually made our corpus closer to a canonical corpus, we have computed the perplexity of the original sentences and of the normalized sentences estimated by a 5-gram Kneser-Ney language model trained on the OpenSubtitles corpus: the normalized version has a perplexity of 2,214 (and 11.60% of its token are OOV) far lower than the original version (with a perplexity of 8,546 and an OOV ratio of 19.60%).

the target can be generated from the 'normalized' source.

In the end, the annotation of this corpus represents 200h of work, comprising an iterative improvement and debugging of the annotations to achieve the corpus' current version. [4]

On the other hand, to reduce the lexical variability and model the impact of UGC's named entities on translation quality, all occurrences of these have been manually mapped to a frequent named entity in the train set, 'Jean' (a French first name). A similar strategy has been proposed by Marton et al. (2010) to reduce the diversity of number in parsing. Considering a 'real' named entity allows us to preserve the structure of the sentence, which would have been compromised by simply erasing NE. These approaches proved to be experimentally adequate, since we can see in the column `named entity` in Table 6.5 that our fully-normalized version using this trick, performs better than its noisy counterpart (the original UGC corpus only keeping named entities occurrences) across our MT models and metrics.

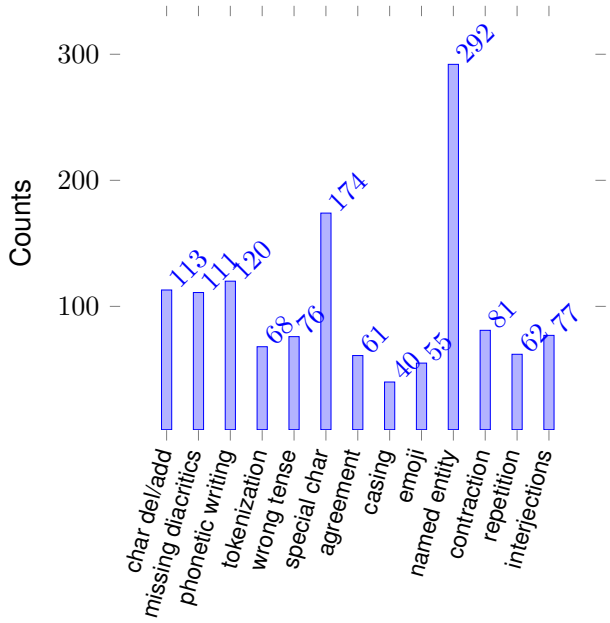


Figure 6.4: Distribution of UGC specificites in `PMUMT`.

The resulting corpus contains more than $1,310$ annotations. On average, each sentence contains 2.8 UGC peculiarities; the average span length of the UGC specificities is 1.27 tokens and its maximal value 17 tokens (a full sentence in upper-case). Figure 6.4 describes the distribution of UGC peculiarities in the corpus. It appears that the most frequent kinds of specificities found in UGC are the presence of named entities and special characters (e.g. @ or

[4]The annotated corpus and code collection can be found in https://github.com/josecar25/PMUMT_annotated_UGC_corpus/

#) or emoticons. Although the least frequent type is the inconsistent case changing, these have the longest span among all the UGC specificities, as some sentences are written completely in uppercase.

Additionally, in Table 6.4, we display some of the annotated examples using their corresponding typology (Table 6.2) and position in the source and reference sentences.

①	src	**JohnDoe389 (10)** qui n'arrive pas **a (2) depasser (2)** 1 a **(2) FlappyBird** ... **ptddddr (12,13)**
	ref	JohnDoe389 who can't score more than 1 at FlappyBird ... lmaooooo
	N. src	Jean qui n'arrive pas à dépasser 1 à Jean ...
	N. ref	Jean who can't score more than 1 at Jean...
②	src	**#CaMeVénèreQuand (6)** le matin **a (2)** 7h on me parle alors que je suis pas encore **réveiller. (5)**
	ref	#ItAnnoysMeWhen in the moring at 7 am someone talks to me although I didn't wake up yet.
	N. src	le matin à 7h on me parle alors que je suis pas encore réveillé.
	N. ref	in the moring at 7 am someone talks to me although I didn't wake up yet.
③	src	vu sa **tete (2) c (3)** normal **kon (3) est (3) jms (11) parler (5)** d'elle !
	ref	in light of her face it's normal no one ever spoke about her!
	N. src	vu sa tête c'est normal qu'on a jamais parlé d'elle !
	N. ref	in light of her face it's normal no one ever spoke about her!
④	src	y a ma cousine qui **joue a (2)** flappy bird **(10) mdrrrrrrrrrr (12, 13) elle et plus nuuul (12,7) que moi**
	ref	my cousin plays flappy bird looooooooooool she's more hopeless than me
	N. src	y a ma cousine qui joue à Jean Jean elle et plus nulle que moi
	N. ref	my cousin plays Jean Jean she's more hopeless than me

Table 6.4: Examples from our annotated noisy UGC corpus. Source sentences have been annotated with UGC specificities of Table 6.2 (in blue) according to their numerical code. For each example, the original source and reference (*src* and *ref*) and their corresponding normalized version (*N. src* and *N. ref*) are shown.

Controlling the Number of Specificities per Sentence Thanks to the alignment between the UGC specificities in a sentence and its normalized form, it is possible to create a normalized version of each source sentence that is closer to canonical French. Comparing the predictions of an NLP system, taking either the normalized sentences or the original non-canonical sentences as input, allows us to measure the impact of UGC on this system. However, it is impossible to perform a fine-grained analysis in which, for example, the impact of different types of specificities are compared, since UGC sentences generally contain several specificities of different types and the interactions between them cannot be easily neutralized: in our corpus, there are only 68 sentences in which a single kind of specificities occur, i.e. 17% of the original sentences.

That is why, a second version of the corpus has been automatically generated to aid analyzing the interactions between the UGC specificities in a sentence: by substituting only some of the

span we have annotated, it is possible to create corpora in which the number and the kind of specificities present in each sentence is tightly controlled: we can, for instance, generate a corpus in which sentences have at least two specificities of the 1st, 2nd and 3rd kind. In this framework, each original sentence can be (partially) rewritten into as many sentences as there are UGC specificities in it.

This possibility of partial substitution greatly reduces the amount of data to be annotated for the analyses: instead of having to annotate a large amount of data to find enough sentences fulfilling the requested criteria, the framework is capable of generating these sentences from our original annotation of 400 sentences. For instance, by substituting all spans but one in our original corpus, we have generated a corpus of 1,282 sentences each containing exactly one UGC specificity and by substituting all spans but two, a corpus of $1,176$ sentences containing two different specificities. We will see in Section 6.5.1 the importance of controlling the number of specificities per sentence to unravel the complex interactions between them.

6.5.1 Impact of UGC specificities on translation quality

	original	normalized
seq2seq	25.8	32.4
char2char	24.1	30.5
Transformer	28.6	33.6

Figure 6.5: BLEU scores on the original and normalized source sentences of the PMUMT corpus.

We have used the PMUMT corpus to evaluate the impact of UGC peculiarities on translation quality: we have reported in Table 6.5 the BLEU scores achieved by the considered systems on both the 400 original sentences and the 400 normalized sentences. As expected, translations of normalized sentences, that are more similar to the training data, are of better quality than translations of original (noisy) sentences: the BLEU scores achieved when translating normalized UGC content are close to those obtained on the in-domain test-set.

For all systems, considering the non-canonical original sentences results in a drop in translation quality of the same order of magnitude, which shows that, even if these models build sentence representations from completely different information, the presence of UGC peculiarities have a similar impact on all of them.

Individual UGC errors To get a more precise picture of the impact of UGC on translation quality, we have computed, for each kind of peculiarities, the BLEU scores achieved on the corpus built to contain only this peculiarity and the BLEU score computed on the 'normalized' version of the same sentences. Table 6.5 reports the ratio between these two scores. Additionally, we have calculated the translation scores using more metrics, namely, CHRF and MULTI-BLEU-PERL-DETOK (MB) along with their 95% confidence interval. This is done in order to ensure that we have a large-enough data collection to obtain a stable corpus-level metric, such as BLEU.

	Metric	char del/add	missing diacritics	phonetic writing	tokenization	wrong tense	special char	agreement	casing	emoji	named entity	contraction	repetition	interjections
s2s	MB	0.78 (25.3)	0.94 (31.1)	0.92 (23.2)	0.95 (30.6)	0.98 (28.6)	0.83 (24.8)	0.95 (25.1)	0.77 (26.7)	0.87 (29.6)	0.87 (30.9)	0.91 (28.4)	0.83 (29.9)	0.90 (26.5)
	chrF	0.93 (46.7)	0.97 (53.1)	0.89 (43.1)	0.95 (50.9)	0.99 (50.6)	0.91 (44.3)	0.98 (49.3)	0.76 (40.2)	0.94 (51.1)	0.94 (51.7)	0.93 (47.3)	0.93 (47.3)	0.96 (48.0)
	SB	0.80 (28.7)	0.95 (33.9)	0.93 (27.3)	0.96 (30.8))	0.94 (30.7)	0.88 (26.1)	0.95 (27.1)	0.75 (27.7)	0.91 (31.0)	0.86 (31.7)	0.95 (30.8)	0.90 (30.2)	0.93 (29.2)
c2c	MB	1.00 (29.5)	1.00 (27.4)	0.85 (22.5)	0.99 (29.7)	0.97 (26.9)	0.80 (23.5)	0.97 (25.5)	0.91 (27.7)	0.83 (25.1)	0.95 (31.7)	0.88 (26.6)	0.93 (28.0)	0.91 (25.7)
	chrF	0.99 (48.5)	1.00 (50.6)	0.92 (44.8)	0.95 (50.1)	0.99 (49.1)	0.84 (44.0)	0.98 (49.9)	0.78 (40.6)	0.93 (49.5)	0.95 (51.6)	0.92 (48.8)	0.90 (47.8)	0.95 (49.7)
	SB	0.99 (32.5)	0.99 (29.6)	0.86 (25.2)	1.00 (31.9)	0.97 (28.8)	0.81 (24.6)	0.96 (28.9)	0.86 (28.0)	0.83 (26.2)	0.94 (32.7)	0.91 (30.4)	0.95 (26.2)	0.91 (28.7)
TX	MB	0.96 (30.3)	1.01 (33.0)	0.98 (33.2)	0.98 (31.5)	1.01 (31.6)	0.90 (28.4)	0.97 (31.4)	0.98 (25.8)	0.72 (26.7)	1.06 (35.7)	0.90 (28.4)	0.81 (25.9)	0.83 (27.0)
	chrF	0.95 (48.2)	1.00 (52.3)	0.98 (46.6)	0.99 (51.0)	1.01 (52.4)	0.93 (46.5)	0.97 (50.9)	0.80 (30.7)	0.88 (49.1)	1.00 (52.6)	0.93 (48.9)	0.87 (46.2)	0.92 (46.2)
	SB	0.98 (35.3)	1.02 (34.0)	1.03 (33.2)	0.98 (32.9)	1.02 (33.7)	0.92 (29.2)	0.97 (33.8)	0.90 (26.9)	0.75 (28.3)	0.99 (35.4)	0.93 (31.1)	0.85 (26.9)	0.86 (30.2)
CI Err.	(E-3)	4.5 (0.17)	1.5 (0.13)	2.7 (0.11)	2.6 (0.17)	2.4 (0.15)	1.8 (0.12)	1.7 (0.23)	5.7 (0.30)	3.0 (0.23)	2.2 (0.11)	2.2 (0.16)	2.5 (0.24)	3.1 (0.22)

Table 6.5: BLEU score ratios between pairs of noisy and normalized sets of sentences, containing only one UGC specificity. BLEU scores on noisy sets are shown in parenthesis. *Three different metrics are shown for comparison:* MULTIBLEU-DETOK.PERL *(MB) ,* CHRF *and* SACREBLEU *(SB).* Error for 95% confidence intervals (CI Err).

The impact of a given kind of UGC specificity on translation quality is very different from one system to another: It appears that the source sentences representation that MT systems learn to construct are not sensitive to the same kind of noise or errors in the source sentence, and even seem to be complementary. For instance, inconsistent casing strongly penalizes the `seq2seq` model but has only a limited impact on the `char2char` model (with a ratio of, respectively, 0.75 and 0.86 BLEU scores). In contrast, the presence of characters specific to online conversation, such as @ or # results in a substantial decrease in translation quality for

`char2char` with a BLEU ratio of 0.81 , but has less impact for `seq2seq` or `Transformer` (with a ratio of, respectively, 0.88 and 0.92) , suggesting that character-based models are not able to properly model characters that hardly appear in the training set.

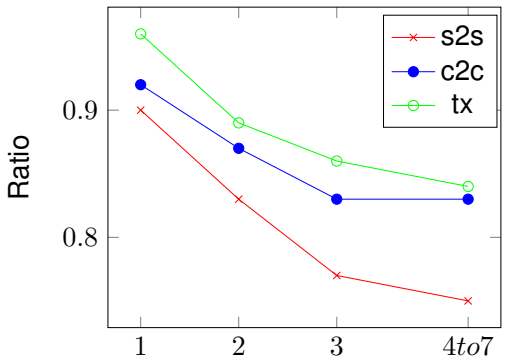
Interestingly, the `Transformer` model appears to be very robust to a wide array of UGC peculiarities, even if it was not designed specifically to handle noisy input: in particular, the presence of named entities, spelling errors (i.e. substitution, deletion, or insertion of letters), agreement error (of verb tense or in gender and number) as well as `tokenization` errors hardly hurt translation quality. Similarly, the `char2char` model succeeds in correctly translating sentences with letter addition or suppression, showing that the model actually manages to learn sentence representations that are robust to spelling errors, even if such errors are not present at training time. This result contrasts with the conclusion drawn by Belinkov and Bisk (2018) on artificially noisy data. However, it fails to capture regularities that are expressed in longer spans (e.g. in named entities).

It is worth noting that `missing diacritics`, `phonetic writing` and `wrong tense` categories give a counter-intuitive result, with a ratio larger than 1.0 (this value being within the confidence interval for the latter), implying that noisy inputs for these UGC specificities seem easier to translate for the `Transformer`. To further investigate this, we have evaluated these corpus partitions again using complementary metrics (discussed in Section 2.4.2), namely, MULTI-BLEU-DETOK.PERL (Koehn et al. 2007a) and CHRF (Popović 2015) in order to recalculate the results in question. In this regard, in Table 6.5 it can be seen that at least one of the metric results in a ratio larger than 1.0, to which we attribute a difference between both BLEU pieces of software.

Combination of specificities Table 6.6b also shows that UGC specificities seem to have less impact when sentences are selected to contain a single specificity: for instance, the `char2char` has an average ratio of 0.942 on the corpora used to compute the scores reported in Table 6.5, while its ratio is 0.79 on the original corpus. To better understand the impact of combinations of UGC peculiarities on translation quality, Table 6.6b reports the ratios between the BLEU scores computed on the translation of a corpus in which there are exactly N different UGC peculiarities in a sentence and on the translation of the normalized version of these sentences. It appears that for all our systems' translation quality decreases linearly with the number of

specificities, suggesting that the impacts of the different specificities are independent of each other. Surprisingly enough, the gap between the `char2char` and `Transformer` is getting smaller with the number of specificities in each sentence.

It can be seen that, initially, the ratios decrease quickly for the three models and, as there are 3 or more UGC specificities, they get to a plateau (plotted in Figure 6.6a). This suggests that the drop in performance caused by such occurrences is substantially more important as soon as noise starts appearing, and their aggregated effect does not increase linearly. Regarding the systems' performance on the normalized versions, we can see an overall downtrend for `char2char` and `Transformer`, probably due to a bias caused by a higher number of UGC specificities being contained, naturally, by longer sentences. Another important factor and a limitation of our data augmentation approach, is that there is an increasingly constrained sentence evaluation diversity, since phrases with a high number of UGC idiosyncrasies are relatively rarer and each original `PFSMB` sentence can generate more augmented samples, proportion is given by the 'Augment ratio' in the table.



(a)

	1	2	3	4+
# sents.	1,306	1,776	1,439	1,089
Augment ratio	3.26	5.82	8.51	13.04
s2s	0.90 (30.1)	0.83 (27.0)	0.77 (24.2)	0.75 (23.2)
c2c	0.92 (29.5)	0.87 (26.6)	0.83 (24.3)	0.83 (23.2)
TX	0.96 (32.8)	0.89 (30.0)	0.86 (28.3)	0.84 (26.5)

(b)

Figure 6.6: (a) Noisy/Clean BLEU scores' ratios for an accumulated number of UGC specificities present per sentence for each model, corresponding to the results in Table 6.6b. The *4to7* bin groups more than 4 types to provide a larger subcorpus, which weighted average is 4.34 UGC specificities per sentence. (b) BLEU score ratio between pairs of normalized and noisy sentences containing N specificities. BLEU scores on noisy sentences are shown in parenthesis.

Additionally, it can be noticed that scores over the normalized sentences are mixed but their weighted averages by the number of phrases in each column, confirm the `Transformer`, `seq2seq` and `char2char` observed performance trend in the test sets in Table 6.1. Interestingly, in Table 6.5, where we present the quotient between noisy and normalized versions per UGC peculiarities within a 95% confidence interval, we can see that errors `1`

- letter change/deletion/addition, 2 - diacritics omission/error and 5 - wrong verb tense for char2char, have a very close ration to 1.0, especially when compared to their BPE-based counterpart with the same architecture, seq2seq. Examples of the impact of such occurrences on MT performance are displayed in elements ①, ② and ③ in Table 6.6. On the other hand, 8 - inconsistent casing seems to impact the performance of char2char the least compared to the BPE seq2seq, suggesting higher robustness of the latter to this type of UGC particularity, for which examples ⑨ and ⑩ demonstrate a noticeable difference in performance. These two observations are in line with the findings of (Niu et al. 2020), which found subword-regularized NMT to be more robust to misspellings in general (here divided in the former 3 UGC specificities evaluated using character-level regularization of char2char) and case-changing. Additionally, we can also notice some advantage of using char2char when for the error 4 - faulty tokenization over seq2seq, but it is surpassed by the BPE-based Transformer, as can be seen in example ⑤ and ⑥ in Table 6.6.

Interestingly, on the other hand, we can notice that error 6 - special character and 9 - emojis/emoticons (usually also out-of-vocabulary) caused the highest relative performance drop when comparing char2char to seq2seq. Such observation is illustrated in example ❶, where we notice the difficulty of the char2char model to output such domain-related characters. This motivates our efforts in the following sections to make char2char more robust to these occurrences, which we address as out-of-vocabulary characters (charOOVs) from now on.

6.6 Improving robustness by learning to manage OOV characters

This set of experiments is motivated by observing that, even though character-level NMT systems are often praised by their word-level open vocabulary, we observed the downside that characters not presented (charOOVs) or underrepresented in the training data impose challenges to these architectures. In this respect, we hypothesize that producing the special <UNK> token, to be replaced by a source-side token during post-processing (as discussed in Section 2.3) can be a better alternative to avoid incorrect translation for some sequences that are underrepresented or not contained at all in the training data. In our framework, this aspect is especially important since the robust NMT systems we aim to conceive for noisy social media text can contain any kind of charOOVs (notably an ever-growing set of emojis) only limited by the users' will.

		Letter deletion/addition/change
①	src	j'arrive pas à **boir normalemen**
	norm	j'arrive pas à boire normalement
	ref	I can't **drink normally**
	s2s	I can't **drink normal.**
	c2c	I can't **drink normally.**
	Tx	I can't **drink normal men.**
②	src	Je conseille à **toux** ceux qui ont l'esprit disons, un peu fermé de regarder sur les "Français d'origine contrôlée"
	norm	Je conseille à tous ceux qui ont l'esprit disons, un peu fermé de regarder sur les "Français d'origine contrôlée"
	ref	I advise **everyone** with a, let's say a little narrow mind to watch about the "Français d'origine contrôlée"
	s2s	I suggest **cough** those who have minds say, a little closed to look at the "frances of controlled origins"
	c2c	I counsel **those** who have the mind, a little close to looking at the French original controlled original controlled.
	Tx	I advise **anyone** with a mind, say, a little closed to look at the controlled French.
⑤	src	le côté suis **tro cool** au quotidien et je **relach** tout **quan** j'ai bu
	norm	les gens qui m'aiment me détestent quand j'ai bu
	ref	my side **very cool** in everyday life and **loosen** everything **when** I've been drinking
	s2s	I've been drinking all the time and I've been drinking everything **quan** I've been drinking.
	c2c	the side of the daily **cool** side and **relacing** everything **when** I've been drinking
	Tx	I'm the **cool** side. I'm the cool one.
		Tokenization
⑥	src	**J'sais pas vous**, mais de voir la joie des grands joueurs comme Zlatan, Motta, Verratti je trouve ça magnifique
	norm	Je sais pas vous, mais de voir la joie des grands joueurs comme Jean, Jean, Jean je trouve ça magnifique
	ref	**I don't know about you**, but seeing the joy of great players like Zlatan, Motta, Verratti I think it's wonderful
	s2s	**I don't know you**, but seeing the joy of the great players like Zlatan, Motta, Verratti, I think it's beautiful.
	c2c	**I don't know about you**, but to see the joy of great players like Zlatan, Motta, Varratt, I think it's beautiful.
	Tx	**I don't know about you**, but seeing the joy of big players like Zlatan, Motta, Verratti, I think it's beautiful.
⑦	src	pendant que vous me laissez en chien à l'atelier mon score de flappy bird fait que **d augmenter**
	norm	pendant que vous me laissez en chien à l'atelier mon score de Jean fait que d'augmenter
	ref	while you're bailing out on me at the workshop my flappy bird score **is just increasing**
	s2s	when you leave me as a dog when you leave me as a dog at the workshop.
	c2c	while you leave me dog at the workshop my flappy bird score **is that increasing**
	Tx	while you leave me as a dog at the workshop my flappy bird score **is just up.**
⑧	src	il **ma** dit que c'était normal aussi et que ça allait redescendre,
	norm	il m'a dit que c'était normal aussi et que ça allait redescendre,
	ref	**he told me** it was normal too and that it would come down,
	s2s	**He said it** was normal, too, and it was going to go down,
	c2c	**He told me it** was normal, too, and it was going back,
	Tx	**He told me it** was normal, too, and it was gonna come down,
		Inconsistent casing
⑨	src	Jean **DANS VOS YEUX**
	norm	Jean dans vos yeux
	ref	Jean **IN YOUR EYES**
	s2s	Jean **D in VOSY**
	c2c	Jean **in your eyes**
	Tx	Jean **in your eyes**
⑩	src	**JE VIENS DE VOIR** Jean **ET** Jean **JE PEUX PLUS**
	norm	Je viens de voir Jean je peux plus
	ref	I JUST WATCHED Jean AND Jean CAN'T TAKE IT
	s2s	I'm going to kill Jean and Jean I can't believe it.
	c2c	I'm here to see Jean And Jean I can no longer.
	Tx	I just saw Jean and Jean again.
		Domain-specific characters and emojis
❶	src	Avec mes magnifiques jumeaux Jean et Jean **@maxcarver @Charlie_Carver** Λ
	norm	Avec mes magnifiques jumeaux Jean et Jean
	ref	With my wonderful twins Jean and Jean **@maxcarver @Charlie_Carver** Λ
	s2s	with my gorgeous Jean and Jean **@maxarver @Carlie_Carver @Carlie_Carver #**
	c2c	with my beautiful Jean twins, Jean Jean and Jean **Charlier Charlier Carver.**
	Tx	with my beautiful twins Jean and Jean **imexcarver Charlie_Charver @Charver**

Table 6.6: Examples from our noisy UGC corpus showing the `Transformer`, `char2char` and `seq2seq` predictions. Present UGC specificities of Table 6.2 (in blue) are marked in bold.

We have noticed that the special <UNK> token management and generation are highly affected by the training vocabulary size for character-based models, i.e. if there are not enough characters excluded from the vocabulary (and consequently, replaced by <UNK>) at train time, the translation system does not learn how to correctly manage such tokens when present during test time. For the sake of this observation, we trained character-based systems with different vocabulary sizes, for which were observed <UNK> being generated during evaluation only when retaining 90 characters or fewer. In terms of configuration, our new char2char systems have 6 vocabulary size variations: 90, 85, 80, 75, 70 and 65. In this way, we hypothesize that the choice of optimal vocabulary size, often overlooked in the NMT character-based literature, can help to improve generalization over out-of-domain (OOD) test sets, while keeping optimal performance on in-domain test evaluation. Furthermore, regarding our two noisy UGC test sets, as we previously mentioned, the PFSMB contains an especially high overlap between source and reference compared to the MTNT, which serve us to evaluate whether our strategy can benefit the most from copying from the source without compromising translation of noisy texts that, change much more between source and reference.

vocab. size	PFSMB	MTNT	News	Open
90	23.9	25.8	18.7	26.6
85	23.9	25.3	**19.9**	**26.9**
80	23.9	25.8	18.3	26.6
75	24.5	**25.9**	17.8	26.3
70	**24.6**	25.4	17.8	26.3
65	22.7	25.5	18.0	26.4

Table 6.7: BLEU results for MT of char2char with reduced vocabulary size.

	PFSMB blind	MTNT blind
Transformer	19.0	**25.0**
seq2seq	**22.1**	20.4
char2char-base	17.8	20.9
+vocab-75	18.3	24.0
+vocab-70	18.7	22.8

Table 6.9: BLEU results for reduced-vocabulary MT systems.

To follow, we report results for variations of the base char2char (Lee et al. 2017) model (originally with 302 characters kept in the vocabulary) with considerably less number of characters

in an effort to correctly train the model to produce <UNK> when needed. In Table 6.7 we can see that the best char2char for our noisy UGC test sets are those with a similar vocabulary size of 70 and 75, which improves the performance of the base char2char in Table 6.1 by +0.8 and +0.6 BLEU points on the PFSMB and MTNT test sets respectively. Furthermore, these models do not lose any performance on in-domain evaluation, having the same or higher BLEU performance for newstest'14 and OpenSubTest clean test set for all the reduced vocabulary size systems. It is also worth noticing that the vocab-85 system also improves the newstest'14 by +2.1 BLEU points, which is a different domain to the training corpus (OpenSubtitles).

Additionally, in Table 6.9, we show results for the noisy UGC blind tests, for which we can observe a BLEU performance improvement on both tests, outperforming the seq2seq model with <UNK> replacement. In turn, with these results, we noticed that the BPE-based Transformer continues to be an upper-bound of performance for the character-base models considered, although the gaps have been reduced.

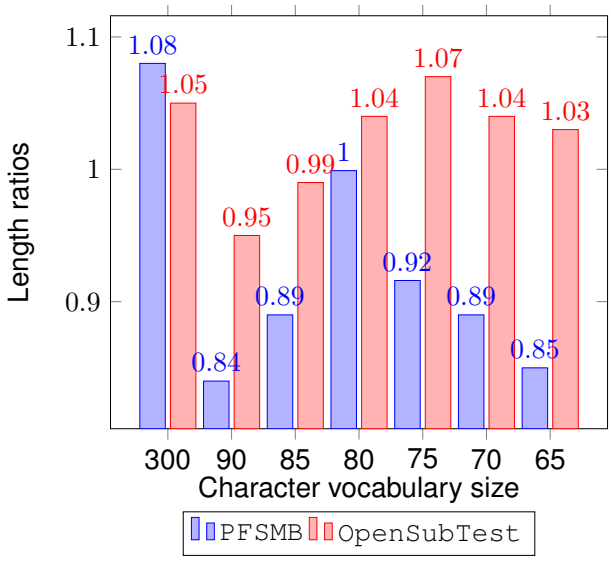


Figure 6.7: Reference/hypothesis length ratios for different vocabulary sizes.

In order to study the occurrences of abnormally short (dropped translations) or long predictions (over-translation), we display the length ratio between predictions and references for the noise UGC test set, PFSMB, and the clean corpus OpenSubTest in Figure 6.7. It can be seen that the base char2char model with 300 characters in the vocabulary incurs in 8% and 5% over-translation for such test sets, respectively. When we vary the number of characters retained in the vocabulary, we succeeded to control such behavior. However, for other sub-optimal

vocabulary sizes (90, 75, 70, and 65), dropped translations were present with up to 16% and 5% of shortening on the predictions with respect to the references on PFSMB and newstest'14, respectively. It is interesting to note that the variability between the lowest and the highest prediction length is considerably bigger for the noisy UGC test. These results point to the fact that an optimal choice of the vocabulary size can alleviate excess tokens production.

6.7 Phoneme2Char translation: a character-based phonemic MT

Our method presented in Chapter 5 cannot resolve phonetic writing errors that are segmented on multiple source tokens, and that cannot generate more than one normalization tokens (we have only considered single words when looking for phonetically-closed elements in the dictionary described in Section 5.1, otherwise, the computation time becomes prohibitive). With this in mind, we intend to overcome such limitations by exploring the possibility of training MT systems that can directly translate from the French phonetization form of sentences to English text. By doing so without using an explicit tokenization, segmentation is either determined using phoneme-based BPE units or 1-grams for comparison purposes, producing NMT systems that fully perform the complete phonetization-normalization pipeline.

Phoneticizing and translating For the data used, we phoneticized the French source-side of the OpenSubtitles dataset training, development and test corpora used to train and evaluate its respective MT baseline in this chapter, using the G2P phonetizer, as seen in Section 5.1. Characters and sequences that are not phonetizable were detected using a regular expression and kept unchanged in their corresponding relative position with respect to neighbor phonemes. To keep the same case-sensitive MT comparison configuration, the original pre-phonetization casing is encoded in the source phoneme sequence to learn to generate the corresponding target casing, by using two different placeholders ("*<T>*" to mark capital letters and "*<U>*" for fully-uppercased tokens) to keep the case information in the phoneticized version of the source sentences leveraging on Factored NMT (Garcia-Martinez et al. 2016), corresponding to the inline casing approach used in Berard et al. (2019c).

Three models are compared: the Transformer model, either with a 16K BPE tokenization over the phonemes of IPA character-level segmentation; and the fully character-level char2char

model (Lee et al. 2017), that we refer to as `phon2char`, standing for the phoneticized source version. The latter, is supposed to process the input as a stream of characters, which segment (n-grams, possibly overlapped) embeddings are extracted end-to-end using a CNN block, as seen in Chapter 2. Additionally, we also show the MT results using a `Transformer` model to compare how self-attention NMT systems would perform on this phoneme-to-text task approach. For the `Transformer`, we have chosen to explore using an IPA-level character segmentation and a 16K BPE tokenization version.

Additionally, we have experimented using two versions of the phoneticized source corpus for training: either keeping the spaces between original words after phonetization to take advantage of the original tokenization information, or removing such spaces and leaving only the phonemes. We found that the latter consistently outperformed the former, probably because of inter-word liaisons in French pronunciation, however, since leaving the spaces leads to inherently larger source sequences, and we do not have access to enough UGC data to produce a control corpus of both approaches with comparable sequence lengths, we cannot rule out the impact of sequence length, nor it is in the scope of this work. These experiences, thus, only served the exploratory purposes of identifying the best pre-processing in order to produce the most performing phoneme-to-text MT systems.

Results In Table 6.8, the results of using the alternative method discussed previously. It is very interesting to notice that. for this task, the `char2char` model achieved a notable performance for UGC and OOD test (`newstest'14`) compared to the `Transformer`, which contrasts to our previous text-to-text MT experimental results. However, we can see that the latter performs better in in-domain translation, which points to low robustness properties on this MT task. In the second place, it can be noted that the finer-grained IPA-level `Transformer`, is systematically outperformed by its BPE tokenization counterpart, probably because BPE helps the model to disambiguate between very similar pronunciations by containing larger phoneme tokens.

Finally, we can notice that using only the phoneticized source information is detrimental to MT, at least in overall translation quality compared to our phoneticized correction systems in Chapter 5, which achieves +5.1 and +3.1 BLEU over in-domain (`OpenSubTest`) and UGC (`PFSMB`) test sets respectively for the `char2char` model. This suggests that conceiving methods that exploit both phoneticized and original text information are worth it over purely phoneme-

to-text training data, and justifies up to some extent the processing over-head needed for our normalization pipeline in Chapter 5.

Small Training Corpus				
	PFSMB	MTNT	News	Open
phon2char	20.7	20.1	13.5	21.7†
Transformer				
16K Phon. BPE	10.4	12.1	4.4	22.5†
IPA-level	5.2	7.3	2.2	18.8†

Figure 6.8: BLEU score results for our three phoneme-to-text models on clean and noisy test sets. The best result for each test set is marked in bold, in-domain scores with a dag.

6.8 Robustness of our proposed character-based NMT models

In order to identify and characterize the robustness to phonetic writing of our phon2char model, which was the most performing system in Table 6.8, we evaluate our PMUMT corpus with it and we present these results in Table 6.10 for a single and unique UGC specificity, and in Table 6.9 for all types of UGC confounded only varying the total number of specificities occurrences (from 1 to 4+).

In the first place, in Table 6.10 we can notice that the corresponding MT system (char2char) becomes more robust to phonetic writing specificity, compared to the SB metrics in Chapter 5 Table 5.2, specifically, it increases in +13% (from 0.86 to 0.97), however, the difference in overall translation performance (lower for phon2char as discussed before), make the noisy BLEU score not comparable. Additionally, it is interesting to notice that phon2char is +8% and +4% more robust to contractions and inconsistent casing, respectively. These results suggest that phonetization can help to recover from such UGC specificities by approaching the noise in UGC to the phoneticized form of canonical constructs.

On the other hand, phon2char showed a loss of -17% and -4% for tokenization and named entity, respectively with respect to char2char, which could be explained by losing some of the original word-level segmentation information through the phonetization process for the former and adding confusion to names, some of which are not readily french-phonetizable.

In addition, the specificities repetition and interjections, also suffer from a decreased robustness of roughly -7% comparing phon2char to char2char. This can be ex-

plained due to the fact that phoneticizing such sequences will arguably always map to incorrect and rare phoneme sequences, since such specificities are not meant to be phoneticized.

Finally, regarding the impact of a given number of present UGC specificities, we can notice in Table 6.9 that `phon2char` have the same overall robustness profile of `char2char` when comparing to Table 6.6b, but the former seems more robust to a relatively high number of occurring UGC specificities.

System	char del/add	missing diacritics	phonetic writing	tokenization	wrong tense	special char	agreement	casing	emoji	named entity	contraction	repetition	interjections
phon2char	0.96 (31.3)	1.00 (27.0)	0.97 (27.2)	0.81 (25.5)	0.99 (27.4)	0.79 (22.0)	0.99 (28.4)	0.88 (25.3)	0.79 (22.9)	0.90 (27.4)	0.95 (30.0)	0.88 (25.5)	0.87 (26.9)

Table 6.10: BLEU score ratios between pairs of noisy and normalized sets of sentences for the `phon2char` system, containing only one UGC specificity. BLEU scores on noisy sets are shown in parenthesis.

	1	2	3	4+
# sents.	1,306	1,776	1,439	1,089
phon2char	0.91 (27.0)	0.86 (24.2)	0.83 (22.5)	0.81 (22.3)

Figure 6.9: BLEU score ratio between pairs of normalized and noisy sentences containing N specificities for the `phon2char` system. BLEU scores on noisy sentences are shown in parenthesis.

6.9 Conclusions

In this chapter, we have tested the capacity of convolutional character-based NMT systems to translate noisy UGC content, and we have compared their performance to mainstream BPE-based models. We also developed a novel UGC evaluation framework to identify the advantages and caveats of our proposed models when translating UGC.

By using our proposed evaluation protocols, annotated data, and code, we were able to show that, contrary to what could be expected, this kind of system is very sensitive to UGC idiosyncrasies, and it is especially challenging to translate texts that are productive in terms of new forms, new structures, or new domains, although we did notice a generalization advantage over BPE subword units NMT for faulty or missing diacritization and letter change (to constitute a misspelling), inconsistent casing, and graphemic and punctuation repetition.

We have also investigated the effects and caveats of choosing different training vocabulary sizes in order to train character-based NMT systems to correctly treat `charOOVs` and rare characters during UGC evaluation, for which the results strongly point to the importance of carefully choosing the number of characters kept in the vocabulary set. Additionally, this aspect has a considerable effect on the length of the predictions and is also one factor that determines abnormal behavior. In this respect, our analysis concludes that reducing the training vocabulary and looking for its optimal size can help training more robust character-based NMT systems, which process `charOOVs` that can be present on OOD evaluation scenarios and real-world MT applications.

Finally, we combine our new character-based models with phonetization information of the source tokens to propose other methods to cope with UGC, which serves a twofold purpose: as a phoneme-based baseline that allowed us to quantify the performance of our phonemic MT systems in Chapter 5 and justify the extra computing overhead needed; and it also constituted the focus of our evaluation procedure to identify its robustness capabilities beyond phonetic writing.

Chapter 7

Variational inference methods for UGC

The previous chapters have highlighted the difficulties raised by the translation of UGCs and the need to consider a zero-shot scenario in which no parallel data is used during training. We have also described our first experiments to improve the robustness of translation systems to a well-identified UGC specificity. In this chapter, we describe a new contribution of our work aiming at using latent variables models to build better text representations, robust to the whole set of variations that can be found in UGC.

To address the problem raised by OOD texts, an increasing number of works explore the possibility to combine deep learning with latent variable (LV) models: the latter are indeed able to capture underlying structure information and to model unobserved phenomena. The combination of LV models with neural networks was shown to increase performance in several NLP tasks (Kim et al. 2018). In this work, we focus on a specific LV model for MT, Variational NMT (VNMT), introduced by (Zhang et al. 2016) which has been reported to have good performance and interesting adaptability properties (Przystupa 2020; Xiao et al. 2020).

The goal of the work described in this chapter is twofold. First, we aim to evaluate the performance of VNMT when translating French social-media noisy UGC, a kind of OOD texts that has never been considered before in the VNMT literature. We hypothesize that, by leveraging on Variational NMT, latent models can build more robust representations able to represent OOD observations that are symptomatic of noisy UGC and automatically map them to in-distribution instances, which can be more easily translated.

Second, to account for the diversity of UGC phenomena, we introduce a new extension of VNMT that relies on Mixture Density Networks (Bishop 1994) and Normalizing Flows (Rezende

and Mohamed 2015). Intuitively, each mixture component extracts an independent latent space to represent the source sentence and can potentially learn different multi-modal regularization distributions, more robust to noise and capable of better translating UGC.

At the end, our contributions can be summarized as follows:

- we study the performance, in a zero-shot scenario, of VNMT models and evaluate their capacity to translate French UGC into English;
- we introduce a new model that uses Transformer as the backbone of a variational inference network to produce robust representation of noisy source sentences, and whose results outperform strong VNMT and non-latent baselines when translating UGC in a zero-shot scenario. Specifically, our model demonstrates a high robustness to noise while not impacting in-domain translation performance;
- by probing the learned latent representations, we show the importance of using several latent distributions to model UGC and the positive impact of the ability of VNMT models to discriminate between noisy and regular sentences while maintaining their representation closer in the embedding space;
- we report evidence supporting that our VNMT models act as regularizers to their backbone models, leading to more robust source embeddings that can be later transferred with a relatively high performance gain in our zero-shot UCG translation scenario.

7.1 Background

In this section, we review the methods featured in this chapter. We start by discussing variational approaches and their interest to robust NMT. We then study specific architectures used to improve performance and that we rely upon to build our proposed VNMT model.

7.1.1 Variational Neural Machine Translation

Variational Inference (VI) methods (Kingma and Ba 2015) are generative architectures capable, from a distributional perspective, of modeling the hidden (i.e. *latent*) relations that can be found in data. In a sequence-to-sequence MT tasks, where x and y are respectively the source and target sentences, VNMT assume that there exists a random variable, z (known as the latent

state), modeling the implicit structure (i.e. relations) between the bilingual sentence pairs (Zhang et al. 2016).

In the context of UGC translation, we believe that this latent variable can capture the variations between a source sentence and its canonical, normalized form, recovering its underlying meaning and ensuring that their representations are similar. We explore this hypothesis in Section 7.5 by assessing perturbations caused by UGC specificities.

To make computations tractable, in spite of the latent variable, VI combines a so-called *variational posterior* $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{y})$ that is chosen to approximate the *true* posterior distribution, $p(\boldsymbol{z}|\boldsymbol{x})$, and a neural decoder generative distribution, $p_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{z})$, in charge of generating the translation hypothesis conditioned on the latent variable. Once the family of densities q is chosen, the parameters of the two distributions are jointly estimated to model the output $\boldsymbol{y}$ by looking for the parameters $(\boldsymbol{\theta},\boldsymbol{\phi})$ that minimize the *evidence lower bound* objective function:

$$\begin{aligned}\log p_\theta(\boldsymbol{y}) \geq \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{y})}[\log p_\theta(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{z})] \\ - D_{KL}[q_\phi(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{y})||p(\boldsymbol{z}|\boldsymbol{x})]\end{aligned} \tag{7.1}$$

7.1.2 Normalizing Flows

One of the major caveats of variational methods is that choosing the prior $q(\boldsymbol{z})$ is a complicated process that requires some *a priori* knowledge of the task. In practice, a normal distribution with fixed parameters (generally$\mu = 0.0$ and $\sigma = 1.0$) is often chosen for computational reasons. However, such an assumption can be restrictive when modeling more complex processes.

Regarding this issue, Rezende and Mohamed (2015) propose to enhance variational methods with Normalizing Flows (Tabak and Turner 2013; Tabak and Vanden-Eijnden 2010). A normalizing flows is a series of simple bijective functions automatically chosen to extract a more suitable representation for the task at hand from a random variable.

In MT, normalizing flows were recently used to improve VNMT models: Setiawan et al. (2020) show that using them in an in-domain evaluation setting results in an increase of +1.3 BLEU points on the IWSLT'14 (De-En) and +0.2 BLEU points on the WMT'18 (En-De).

7.1.3 Mixture Density Networks

Mixture Density Networks (MDN) are another interesting generalization of variational encoding for modeling UGC. In MDN, the posterior distribution $p(\boldsymbol{z}|\boldsymbol{x})$ is no longer approximated by a single variational posterior $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{y})$ but by a linear combination of variational posteriors $\tilde{q}_m(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{y})$:

$$q(\boldsymbol{z}|\boldsymbol{x}) = \sum_{m=1}^{M} \alpha_m(\boldsymbol{x},\boldsymbol{y}) \cdot \tilde{q}_m(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{y}) \tag{7.2}$$

where α_m are known as the mixing coefficients. Intuitively, an MDN is a combination of M variational encoders. Our intuition is that UGCs contain several different kinds of variations covering very different aspects ranging from morphology to phonemics, including lexicon and sentence structure (see Chapter 3) and it is illusory to hope that a single latent variable will be able to capture all of them. But, with an MDN, it is possible that each component of the encoder is able to model different UGC specificities, allowing us to better model UGC. In the past, MDN has been used to address sequence-to-sequence generative tasks, such as `SketchRNN` (Ha and Eck 2018) and modeling of sequential environment states in reinforcement learning (Ha and Schmidhuber 2018).

7.1.4 Gumbel-Softmax sampling

Regarding the mixing coefficients computation, we also explore the use of a categorical probability distribution, for which probabilities are calculated by the network, such as in Ha and Eck (2018). Unlike theirs, our supervised end-to-end training requires backpropagating the error gradient through the variational network via reparameterized sampling (Kingma and Welling 2014) which poses optimization challenges because of the discrete random variables used as latent vector for categorical distributions. For this reason, we use the reparameterization of this distribution via the Gumbel-softmax sampling (Jang et al. 2017; Maddison et al. 2017), such that, the $\arg\max$ function is approximated by a $\mathrm{softmax}$ and generates the relaxed one-hot encoded samples corresponding to the mixing coefficients:

$$\alpha_m = \frac{\exp(\log(\pi_m) + g_m)/\tau)}{\sum_{j=1}^{M} \exp((\log(\pi_j) + g_j)/\tau)} \tag{7.3}$$

where $g_m ... g_M$ are *i.i.d* sampled from the Gumbel(0,1) distribution (Gumbel 1954; Maddison et al. 2017), π_i is the probability associated to the *m*-th MDN's Gaussian components, jointly generated by neural networks along with the computations of the corresponding parameters (μ_m, σ_m) for $m...M$; and τ is the temperature parameter, which controls variability of the sampling. When $\tau \to 0$, the sampling exhibits a perfectly one-hot encoded output, whereas, conversely, when $\tau \to \inf$, the distribution approaches a uniform one across all the MDN's components.

7.2 Extending variational methods for robust MT

The model we designed in this work adopts a variational encoder-decoder architecture inspired by `SketchRNN` (Section 7.1) that uses an MDN on the decoder's variational network to model multiple and independent continuous generative variational distributions. However, unlike `SketchRNN`, we use a Transformer backbone for the encoder and the decoder and train our model in an end-to-end manner on canonical parallel corpora. In the following, we will first describe the general architecture of our model, denoted `multi-VNMT`, and then detail the encoder and decoder parameters.

7.2.1 General architecture

Figure 7.1a shows the architecture of our model. The source sentence is processed by a standard Transformer encoder, whose output is passed as input to a VI network enhanced with NF to predict a latent representation of the input sentence. This vector and the output of the Transformer encoder's last layer are combined using the gating mechanism of Setiawan et al. (2020).

This combined representation is then fed to the decoder that has a similar architecture: it consists of a "standard" Transformer decoder and an MDN. The latter is sampled to obtain a prediction that will be combined with the Transformer output by (again) a gating mechanism.

The model can be trained in an end-to-end fashion using the "reparameterization trick" (Kingma and Welling 2014). In order to ensure that the estimated standard deviations for the variational posteriors are positive, we used the *softplus* activation function (Zheng et al. 2015), as done in van den Berg et al. (2018)'s implementation. This choice alleviates the possibility of exploding gradients in comparison to the exponential function, often used to serve the same

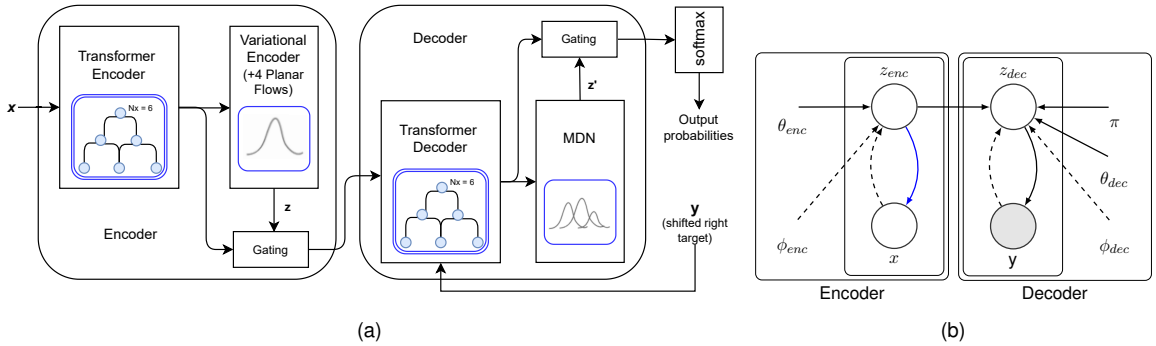


Figure 7.1: (a) `multi-VNMT` architecture overview. (b) Directed graph of our encoder-decoder model variational inference. Dashed lines represent the variational approximation for the posterior distribution, and solid lines stand for the generative models. The blue arrow depicts the generative networks for source-side monolingual reconstruction distribution $p(\boldsymbol{x}|\boldsymbol{z})$.

purpose (Ha and Eck 2018).

We also explored two methods to compute the mixing coefficient in the decoder's MDN: either using $\mathrm{softmax}$ (non-latent approach), or resorting to relaxed categorical variational method that relies on a Gumbel-softmax sampling (Section 3.8).[1]

7.2.2 Encoder

According to our `Transformer Base` baseline architecture from Vaswani et al. (2017), the encoder is composed by a 6-layered `Transformer Base` encoder, which output is feed to a 128-dimensional variational network, that estimates the final latent hidden encoded vector.

In Figure 7.1b, we show the Transformer and variational encoding latent state ($\boldsymbol{z}$) as being estimated ($p(\boldsymbol{z}|\boldsymbol{x})$) approximating the posterior distribution parameters, learned using the reparameterization trick (Kingma and Welling 2014). On the other hand, the blue arrow in the figure, shows how we can introduce a source-side reconstruction loss in order to introduce mono-lingual training generative posterior ($q(\boldsymbol{x}|\boldsymbol{z})$), such as seen applied to VNMT in Zhao et al. (2019), used as regularizer to enforce source information to be efficiently propagated and mitigate posterior collapse. In this work, we also study the impact of this source reconstruction as an accessory module for our proposed VNMT model.

In order to be comparable to the recently introduced `NF-VNMT` (Setiawan et al. 2020), we also report results for our VNMT model extending the encoder's variational inference mechanism with a 4-flows Normalizing Planar Flows (PF) (Rezende and Mohamed 2015). Other autoregressive

[1]The model has been implemented in `OpenNMT` (Klein et al. 2018).

normalizing models, such as Sylvester Flows (van den Berg et al. 2018), are available and could prove interesting for higher capacity. However, we decided to only address PF since they are the simplest solution with comparable performance improvement as other more complex flow models, according to results from Setiawan et al. (2020).

Similarly to `NF-VNMT`, we mix the last Transformer layer output to the latent vectors using a gating mechanism, and a feed-forward network in order to upscale the latent representation dimensionality and match the `Transformer Base` decoder number of dimensions (i.e, from 128 to 512); but unlike this model, we do so for both the encoder and decoder blocks' outputs since we introduce variational networks on both sides.

7.2.3 Decoder

The Transformer decoder's last layer output is passed as input to an 128-component MDN, with trainable parameters ϕ, encoding the mean and standard deviation of each one of these multivariate Gaussian kernels; and π, which contains the probabilities of the categorical distribution that generates the mixing coefficient for each component, as seen in Section 7.1. Concisely, we estimate a series of M posteriors parameterized by $\langle \phi; \pi \rangle$, i.e. $\tilde{q}_m^{\phi;\pi}(z_{dec}|\boldsymbol{x}, \boldsymbol{y}_{1:t-1})$, conditioned via the decoder's Transformer, on both the gated latent encoder's output and the last $t-1$ predicted tokens, $\boldsymbol{y}_{1:t-1}$. To compute the MDN's mixing coefficients, $\alpha_m(\boldsymbol{x}, \boldsymbol{y}_{1:t-1})$, we explore either using a fully-connected layer with a $\mathrm{softmax}$ activation, or the relaxed categorical Gumbel distribution. Both networks computing $\tilde{q}_m$ and α_m are jointly trained in an end-to-end fashion, such that translation loss is minimal for representations sampled from the resulting mixture, obtained according to Equation 7.4. We train the MDN by variational inference using reparameterized sampling, similarly to our variational encoder network.

By using MDN, the posterior distribution of the current decoding step, $p(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{y}_t)$, is no longer approximated by a single variational distribution $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{y}_{1:t-1})$ but by a linear combination of variational posteriors $\tilde{q}_\phi^m(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{y}_{1:t-1})$:

$$p(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{y}_t) = \sum_{m=1}^{M} \alpha_m(\boldsymbol{x}, \boldsymbol{y}_{1:t-1}) \cdot \tilde{q}_m(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{y}_{1:t-1}) \tag{7.4}$$

where α_m are known as the mixing coefficients.

Regarding our choice of architecture, we also explored incorporating Normalizing Flows to

the MDN's components variational inference, which resulted in poor performance and very large number of parameters. We, thus, apply these two methods independently, and we leave their simultaneous use within a single variational network for future works.

7.2.4 Jointly-learned MLM representations

An interesting property of our model is that it is inherently capable of being pre-trained (or jointly trained) as an auto-encoder considering only monolingual data in the source language. This pre-trained model can be later fine-tuned on other MT domains or bootstrapped to be used in other NLP tasks. Indeed, the reconstruction error can be minimized as training objective such that the network predicts one or several masked token in the source sentence.

In this work, we use the MASS approach (Song et al. 2019) as masked language model (MLM) scheme, by introducing a supplementary source reconstruction loss while masking the source tokens to predict.

Using these tools, we explore a semi-supervised approach, as done in Zhao et al. (2019), and performed experiments adding a source-side reconstruction loss term, both for MLM and reconstruction objectives:

$$\mathcal{L}_{mono} = \mathbb{E}_{z\sim q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\log(p_\sigma(\boldsymbol{x}|\boldsymbol{z})] - D_{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x})\|p_\sigma(\boldsymbol{z})) \tag{7.5}$$

This objective is maximized by sampling the approximated posterior distribution ($p_\theta(z|x)$) by the means of variational inference, represented as the blue arrow in Figure 7.1b. Using the MASS MLM auxiliary loss and adding a source reconstruction term to it will, intuitively, guide the latent representation of the source sentence by modeling the inner relationships between its tokens.

In our experiments, we only use the source sentences contained by the training datasets in order to be able to unequivocally assess the advantages of this auxiliary task, ruling out the impact of supplementary monolingual data. However, using additional monolingual corpora is arguably the main interest of this approach.

7.3 Experimental Evaluation

7.3.1 Evaluation Protocol

We have conducted experiments on our VNMT systems and the `NF-VNMT` baseline using the data and protocols discussed in Chapter 3. In this regard, we present an ablation test of our proposed `multi-VNMT` system and show the impact of our architecture's design over MT performance across our different test sets. We also performed experiments by incorporating MLM and reconstruction objective functions, as reviewed in Section 7.1. In addition, we evaluate the impact of temperature τ in order to assess whether the sparsity of mixing coefficients favors a given type of texts among our test sets. Concerning this, as an initial experimental configuration, we chose $\tau = 1.0$, which was selected mainly aiming to avoid artificial gradient scaling during backpropagation, directly caused by this coefficient being relatively larger than 0 in order to introduce variability to the sampling process.

We quantify the robustness of `multi-VNMT` by employing the `PMUMT` evaluation framework discussed in Section 6.5. Finally, we present a series of visualizations and metrics to characterize how `multi-VNMT` behaves when processing UGC during evaluation, in order to give further insights of its robustness capabilities compared to the `NF-VNMT` and `Transformer` baselines.

Training models All systems are trained using a batch size of 4,096 tokens using the Adam optimizer (Kingma and Ba 2015) accumulating gradients every two steps, and the `Noam` learning rate schedule (Vaswani et al. 2017) with 8,000 warm-up steps. Throughout training, learning rate reaches a maximum value of 0.0009 and minimal value of 0.0001. Both encoder and decoder Transformers are trained using 0.1 dropout, and we used 0.1 label smoothing (Szegedy et al. 2016). Training for, at most, 300K training iterations on a single Nvidia V100 took about 50 hours to converge for the `multi-VNMT` models. In order to avoid posterior collapse, we use β_C-VAE (Prokhorov et al. 2019), with values $\beta = 1$ and $C = 0.1$, as done in Setiawan et al. (2020). Additionally, we used a Kullback-Leibler (KL) annealing schedule of 80K iterations, i.e. scaling the KL divergence term in Equation 7.1, which allows the model to mostly focus on the translation loss before starting regularizing the KL divergence.

We have also retrained the `Transformer Base` baseline system presented in previous chapters with the optimal hyperparameters found for the VNMT systems, namely, a peak learning

rate of 1.5 for the `Noam` schedule (Vaswani et al. 2017) (instead of 2.0 for the vanilla version) and training for 400K training steps, as opposed to 200K used before. This was done to keep all the NMT systems comparable, and resulted in a small improvement of the `Transformer Base`, as we report in the following section.

7.3.2 MT scores

Test performances achieved by the NMT systems are reported in Table 7.1 using BLEU (Papineni et al. 2002) and `chrF2` (Popovic 2017), both computed by SACREBLEU (Post 2018) with the '`intl`' tokenization, after detokenizing the systems outputs.

We computed the 95% statistical significance by using a 1,000-samples bootstrapping for both BLEU[2] and `chrF2`[3], as in Koehn (2004). It should first be noted that the performances of the three systems we consider are identical when they are evaluated on in-domain data, whatever the evaluation measure considered (no statistically significant difference between the models). This observation highlights one of the strength of the proposed method: contrary to fine-tuning (arguably the most common method to adapt a system to a new domain) that often hurts performance on in-domain evaluation because of catastrophic forgetting (McCloskey and Cohen 1989), the improvement of the quality of UGCs by the proposed method is not at the expense of the quality of translation of canonical texts.

It also appears that, on out-of-domain text, `multi-VNMT`, the approach proposed in this work, outperforms the standard `Transformer` model as well as the state-of-the-art VNMT model, supporting our hypothesis that considering several variational inference components allows to better capture all the variations that can be found in UGC and will result in improved translation quality. Interestingly, our system also performs better than `Transformer` when evaluated on out-domain canonical data and not only on UGC data. It should be noted, however, that the gains of our model are consistent but small and statistically significant mainly when translation quality is assessed using `chrF2`.

Ablation study To better understand the impact of the different components of our model, we conduct an ablation study, whose results are reported in Table 7.2. Overall, we obtain the best

[2] SACREBLEU signature: *nrefs:1|bs:1000|seed:12345|case:mixed|eff:no|tok:intl|smooth:exp|version:2.1.0*.
[3] SACREBLEU signature: *nrefs:1|bs:1000|seed:12345|case:mixed|eff:yes|nc:6|nw:0|space:no|version:2.1.0*.

BLEU scores across all test sets for the "full" multi-VNMT model.

In Table 7.2 we can notice that, overall, we obtain the best BLEU scores across all test sets for the full multi-VNMT version. As interesting mixed results, we can highlight the cases for static latent representation (*z static*), where instead of sampling from the learned distributions, we retrieve their mean as output, and which showed slightly better BLEU scores when translating the MTNT and newstest'14 test sets, with +1.2 and +0.1 BLEU points improvement, respectively. However, results are inconsistent for UGC test sets and otherwise worse than those of the full model for both in-domain and canonical OOD test sets for our two training configurations. This might be explained by the lack of stochastic perturbations provided by the sampling step during training, leading the model to lose generalization during evaluation.

It is also interesting to note that using a categorical variational version of the mixing coefficients rather than the usual choice of computing them with a $\mathrm{softmax}$ improves translations quality: the latter is only performing better for the newstest'14 test set when training on the OpenSubtitles corpus (π *non-latent*). Following the same trend, the WMT training data configuration also shows improvements when using the Gumbel-Softmax version, for which +0.8 and +0.3 BLEU point increment were obtained for both the PFSMB and MTNT UGC test sets, respectively.

Posterior collapse Comparing multi-VNMT and its ablated version system removing the MDN module, both trained on OpenSubtitles and when evaluating the corresponding in-domain test set (OpenSubTest), we have calculated the average KL divergence of the variational decoder's MDN, which resulted in 0.21 and 0.15, respectively. Performing the same analysis for the WMT training and evaluation configuration, KL divergence resulted in 0.38 for the full multi-VNMT and 0.33 for its version removing the MDN block. These results suggest that our proposed architecture is less prone to suffer from the posterior collapse phenomenon, and this could be explained by the use of several independent posterior distributions when including MDN in our model. This could also explain why, in Table 7.1, our systems employing MDN have an overall higher BLEU results than the aforementioned ablated system where we remove this component.

		WMT				OpenSubtitles				
		PFSMB †	MTNT †	News◇	OpenSubTest	PFSMB †	MTNT †	News	OpenSubTest ◇	# params.
BLEU	Transformer	15.1	21.3	**27.9**	16.4	27.7	28.4	26.4	31.4	69M
	NF-VNMT	15.5	21.4	**27.9**	16.4	28.0	28.9	**26.5**	31.4	72M
	multi-VNMT	**16.0***	**21.8**	**27.9**	**16.7***	**28.4**	**29.2**	26.4	**31.5**	77M
chrF2	Transformer	37.8	45.1	54.4	38.6	46.9	48.3	52.6	48.9	69M
	NF-VNMT	38.3	45.1	**54.6**	38.6	47.6	49.2*	**53.1***	48.9	72M
	multi-VNMT	**38.5***	**45.5**	**54.6**	**39.0***	**47.7***	**49.6***	52.9*	**49.0**	77M

Table 7.1: BLEU and chrF2 test scores for our models. The † symbol indicates the UGC test sets, and ◇ in-domain test sets. *Highest metrics for each test set are in bold; scores significantly better than Transformer ($p < 0.05$) are marked with a *.*

	WMT				OpenSubtitles				
	PFSMB †	MTNT †	News◇	OpenSubTest	PFSMB †	MTNT †	News	OpenSubTest ◇	# params.
multi-VNMT	16.0	**21.8**	27.9	**16.7**	**28.4**	29.2	26.4	**31.5**	77M
π non-latent	15.8	21.0	27.8	16.4	28.1	28.5	**26.6**	31.3	77M
-NF	15.3	21.6	**28.0**	16.5	28.3	28.8	26.1	31.3	76M
z static	**16.5**	20.9	**28.0**	16.4	28.1	**29.3**	26.2	31.4	76M
-MDN	**16.5**	20.9	27.8	16.6	27.7	28.7	26.2	31.3	72M

Table 7.2: BLEU test scores our ablated variants. The † symbol indicates the UGC test sets, and ◇ in-domain test sets.

7.3.3 Impact of source-side monolingual joint training

In Table 7.3 we report results with our proposed multi-VNMT system when using source-side monolingual corpora MLM and reconstruction loss terms. It appears that the result we achieve are quite disappointing: for the two training configurations, the reconstruction term (MonoMLM) allows keeping performances on newstest'14 close to those of the baseline, while having little impact on the OpenSubTest corpus. Results on the UGC test sets show improvements, but these are not consistent and depend on the train set chosen. Although conclusions are difficult to drawn from these results, some UGC translation improvements for OpenSubtitles are interesting and more experimentation in needed, specifically mechanisms and protocols to control the contribution of the reconstruction loss terms, such as adding importance weights to the terms, ablation schedules, and adding extra monolingual data that introduce new information.

	WMT				OpenSubtitles			
	PFSMB †	MTNT†	News◇	OpenTest	PFSMB †	MTNT†	News	OpenTest◇
multi-VNMT	**16.0**	**21.8**	27.9	**16.7**	28.4	**29.2**	**26.4**	31.5
Mono-multi-VNMT	15.8	**21.8**	**28.0**	16.5	**29.3**	28.7	26.2	**31.6**
MLM-multi-VNMT	15.6	21.8	27.9	16.5	28.0	28.9	**26.4**	31.5
MonoMLM-multi-VNMT	15.6	21.6	**28.0**	16.5	27.8	29.0	26.3	31.5

Table 7.3: Best system using MLM and source monolingual variational reconstruction loss.

	in-dom test
NF-VNMT (Z ~ 4-PF) (Setiawan et al. 2020)	36.1†
multi-VNMT (Z ~ 4-PF)	36.3

Table 7.4: Translation performance of our considered VNMT models on the De-En IWSLT'14 experimental setup. The † symbol denotes reported results. They have been computed using tokenized BLEU, as we reproduced the same pre-processing and vocabulary parameters.

7.3.4 Results using a standard experimental setup

In order to assess how multi-VNMT compares to NF-VNMT's results reported in Setiawan et al. (2020), we have recreated[4] their IWSLT'14 De-En configuration with 160K training sentence and 10K source-target joint BPE vocabulary, which in turn was adopted in Edunov et al. (2018) and Wu et al. (2019).

This is an interesting alternative setup with a much smaller train set than the ones we considered in our experiments. This setup allows us to assess whether our findings on the two transformer-based VNMT architectures hold for a different language pair and a "low-resource conditions". In this regard, we report our results for multi-VNMT in this setting in Table 7.4, where it can be seen that our model has on-par (slightly better) results compared to the VNMT baseline.

7.4 Qualitative analysis

In Table 7.5, we show some examples from the PFSMB and MTNT test sets and their translation by different NMT systems. We notice a general trend of multi-VNMT (MTX in the table),

[4]We have re-implemented our models using the Fairseq toolkit (Ott et al. 2019) to keep the same NMT framework. We focus on their system using 4 Planar NF as ours use the same approach.

outperforming the baselines and producing overall longer predictions when rare tokens or letter repetition are present in the source sentence. This is the case for Example ① in which some tokens have an inconsistent case ; Example ② contains repeated characters and words ; Example ③ has an out-of-vocabulary character "•", and Example ④ contains a hashtag and several user mentions both identified by OOD characters ("#" and "@" respectively).

		PFSMB
①	src	OOOOOOOUUUUUUUIIIIIIII ZLATAN IBRAHIMOVIC SIGNE UN DOUBLÉ À LA 90eme MINUTE ET ENVOI LE #PSG EN FINALE!!!!
	ref	YYYYYYYEEEEEEESSSSSSSS ZLATAN IBRAHIMOVIC PERFORMS A DOUBLE AT THE 90th MINUTE AND SENDS THE #PSG TO FINAL!!!! 1-2!!!!
	TX	OOOOOOOOOOUUULL ZLATAN IBRAHIMOVYSIG A DOUBLED IN THE 90th MINUTE
	NF	OOOOOOOOOOUUUIIIZLATANIBRAHIMOVIC SIGNS A DOUBL AT THE 90th MINUTE
	MTX	OOOOOOOOOUUUUIII ZLATAN IBRAIMOV SIGNS A DOUBLED ON THE 90th MINUTE!
②	src	ne spooooooooilez pas teen wolf non non non et non je dis non
	ref	don't spooooooooil teen wolf no no no and no I say no
	TX	Don't spoil it.
	NF	No, no, no, no, no, no, no, no.
	MTX	Don't spooooooooil. Don't. Don't.
③	src	Roman Godfrey a regardé Teen Wolf (2011) • S03E17 Silverfinger et retourne dessiner des ronds sur son gitan préféré.
	ref	Roman Godfrey watched Teen Wolf (2011) • S03E17 Silverfinger and is back drawing circles on his favorite gypsy.
	TX	Roman Godfrey looked at Teen Wolf.
	NF	Roman Godfrey looked at Teen Wolf.
	MTX	Roman Godfrey watched Teen Wolf (2011), this is S03E17 Silverfinger and goes back to drawing a round about his favorite gypsy.
④	src	Vient de perdre une grosse heure a #flappybird cc @JohnDoe533 @JohnDoe534 @JohnDoe535
	ref	Just lost a big hour on #flappybird cc @JohnDoe533 @JohnDoe534 @JohnDoe535
	TX	#Flappybird cc #JohnDoe53 #JohnDoe53 #53 #1
	NF	#Flappybird c@JohnDoe5333)@JohnDoe53@JohnDoe53
	MTX	Just lost a huge hour at #flappybird cc at John Doe53 #John John Doe
		MTNT
⑤	src	Le programme 'hwclock'et le fichier de configuration '/etc/adjtime'servent à cela.
	ref	The program 'hwclock' and the configuration file /etc/adjtime' are used for that.
	TX	Program /hwock and file configuration file /adjtime to that.
	NF	The hwclock program and the configuration file are for this.
	MTX	The program 'hwlock' and file of configuration - and/adjtime are at that.
⑥	src	Chui bourré (Côte du rhône master race) et je viens de recevoir ma paie... Vive l'intérim et l'emploi précaire qui te paie le 15 du mois!
	ref	I'm drunk (Côte du rhône master race) and I just got paid.... Long live temp work and unstable jobs that pay you on the 15th of the month!
	TX	Drunk Chui, and I just got my pay... live the temp and the early job that pays you on the 15th of the month!
	NF	Drunk chui, and I just got my paycheck, and I've got the temp and the pre-job that pays you on the 15th!
	MTX	Drunk pot roast, and I just got my pay... cheers for the temp and the job that pays you on the 15th of the month!
⑦	src	Osti jle cherche depuis un boutte le montrer à une amie un brin plus jeune.
	ref	Goodness I've been looking for it for a while to show a friend a younger strand.
	TX	**I'm looking for it** from a piece of paper to a younger friend.
	NF	Ostij's **been looking for it ever since** we started showing it to a friend who's younger.
	MTX	Osti **I've been looking for a bit** to show it to a younger friend.

Table 7.5: Examples from our noisy UGC corpora showing the `Transformer`, `NF-VNMT` and our model, `multi-VNMT`, predictions. *`NF` and `MTX` stand for the `NF-VNMT` (Setiawan et al. 2020) and `multi-VNMT` VNMT systems, respectively.*

7.4.1 Robustness

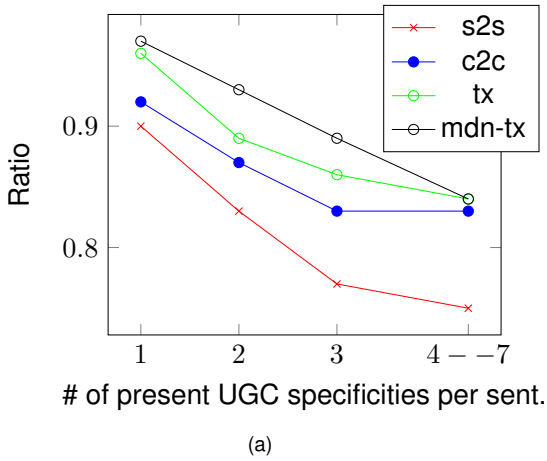
To assess the robustness of our proposed model, we report in Figure 7.2 the noise impact ratio metric (as defined in Section 2.4.3) for each UGC specificity (Table 7.6) and with respect to the number of different UGC specificities in the input sentence (Table 7.2b). It appears that our model is, overall, more robust than all the models we have used so far, notably when OOV and rare character are present in the source (namely, `special char.` and `emojis`. On the other hand, some specificities seem to degrade performance for our architecture, notably `wrong tense`, `agreement` and `repetition`, that might be explained by the several semantically-equivalent sentence versions they represent, and that `multi-VNMT` has problems to disambiguate from. Overall, in Table 7.2b, it can be noticed that `multi-VNMT` is less sensitive to the number of noise occurrences, witnessing a less pronounced impact on performance as more specificities are present in the input. However, for a large quantity of specificities (`4-7` in Table 7.2b), the assessed robustness of our proposed architecture diminishes, matching that of the `Transformer` baseline.

	char del/add	missing diacritics	phonetic writing	tokenization	wrong tense	special char	agreement	casing	emoji	named entity	contraction	repetition	interjections
s2s	0.80 (28.7)	0.95 (33.9)	0.93 (27.3)	0.96 (30.8)	0.94 (30.7)	0.88 (26.1)	0.95 (27.1)	0.75 (27.7)	0.91 (31.0)	0.86 (31.7)	0.95 (30.8)	0.90 (30.2)	0.93 (29.2)
c2c	0.99 (32.5)	0.99 (29.6)	0.86 (25.2)	1.00 (31.9)	0.97 (28.8)	0.81 (24.6)	0.96 (28.9)	0.86 (28.0)	0.83 (26.2)	0.94 (32.7)	0.91 (30.4)	0.95 (26.2)	0.91 (28.7)
TX	0.98 (35.3)	1.02 (34.0)	1.03 (33.2)	0.98 (32.9)	1.02 (33.7)	0.92 (29.2)	0.97 (33.8)	0.90 (26.9)	0.75 (28.3)	0.99 (35.4)	0.93 (31.1)	0.89 (30.8)	0.86 (30.2)
`multi-VNMT`	1.00 (35.5)	0.99 (34.7)	0.95 (34.0)	0.96 (33.1)	1.00 (32.5)	0.92 (30.3)	1.00 (30.2)	0.88 (26.8)	0.83 (30.6)	0.98 (35.1)	0.96 (31.1)	0.86 (28.3)	0.83 (28.2)

Table 7.6: BLEU score ratios between pairs of noisy and normalized sets of sentences, containing only one UGC specificity. BLEU scores on noisy sets are shown in parenthesis.

7.5 Learning representations: where does the magic happen?

In this section, we analyze the neural representations, namely the variational encoder's latent space and source-side embeddings (the very same input layer in the Transformer backbone). We intend to assess how our model behaves when translating UGC in comparison to both our latent and non-latent baselines. By doing so, we also aim to undercover if VNMT indeed learns



(a)

	1	2	3	4+
# sents.	1,306	1,776	1,439	1,089
s2s	0.90 (30.1)	0.83 (27.0)	0.77 (24.2)	0.75 (23.2)
c2c	0.92 (29.5)	0.87 (26.6)	0.83 (24.3)	0.83 (23.2)
TX	0.96 (32.8)	0.89 (30.0)	0.86 (28.3)	0.84 (26.5)
multi-VNMT	0.97 (33.0)	0.93 (30.3)	0.88 (28.4)	0.84 (26.6)

(b) BLEU score ratio between pairs of normalized and noisy sentences containing N specificities. BLEU scores on noisy sentences are shown in parenthesis.

Figure 7.2: Comparison of multi-VNMT's robustness to different number of present UGC specifies.

more robust neural representations, and, confirm that variational approaches act as regularizers and are able to promote robust backbone's representations (embeddings).

7.5.1 Latent space analysis

To assess how VNMT builds more robust learning representations than our baselines, we report the cosine similarity distribution between the representations of the French noisy sentences and their normalized version, taking advantage of the PMUMT introduced in Chapter 6. To obtain these embeddings, we fed the 400 original noisy UGC sentences and their corresponding 400 fully normalized versions to our VNMT baseline, NF-VNMT, and to multi-VNMT.

To measure the perturbations that the model suffers when noise is present in the source, we measure the similarity between the latent representations built by our two VNMT models of the noisy sentences from PMUMT and the representations of their corresponding normalized version. We observe that the average similarity between the representations of multi-VNMT is 0.36 compared to an average similarity of 0.26 for the representations of NF-VNMT, suggesting that the former provides more robust representations of UGC than the former. We also compare the distribution of cosine similarities of both VNMT systems in Figure 7.3, which confirms this trend by showing multi-VNMT's skewed to overall higher similarities between original and normalized sentence pairs.

In Figure 7.4, we show the t-SNE (van der Maaten and Hinton 2008) visualization of both VNMT systems, displaying the latent encoding of noisy and normalized PMUMT sentences.

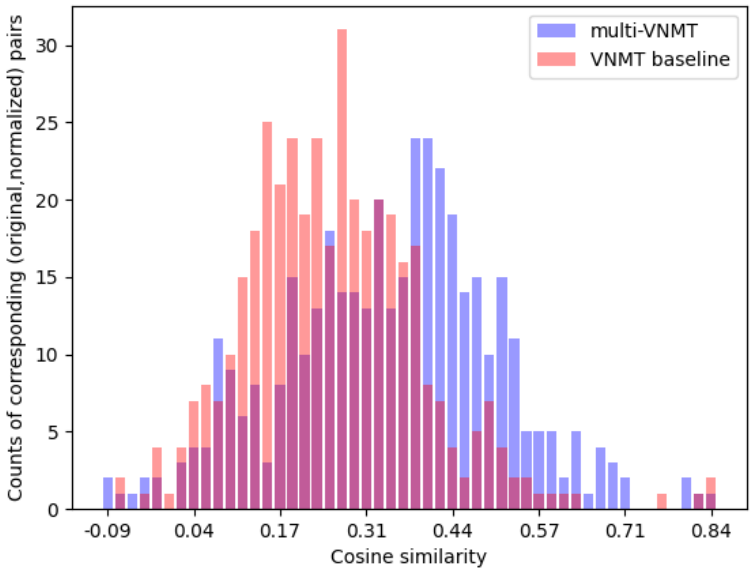
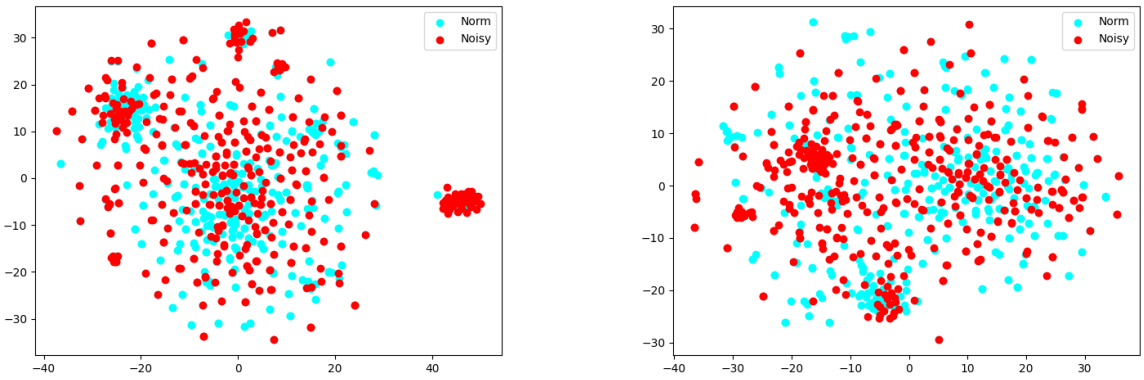


Figure 7.3: Histogram of cosine similarity for corresponding noisy and normalized PMUMT samples in the encoder's latent space of NF-VNMT and multi-VNMT.

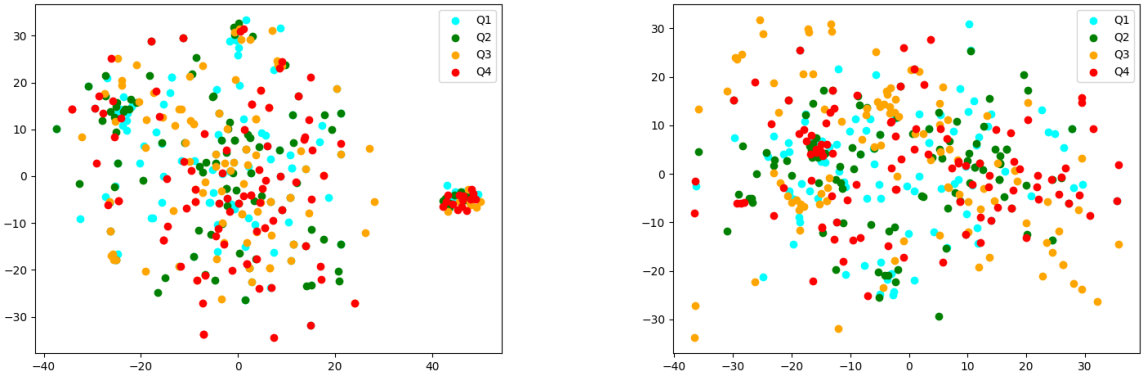
First, we can notice that the NF-VNMT latent representations present a series of outliers when noisy sentences abound, contrary to the multi-VNMT representations, which also have a more compact support. In this set of 43 outlier observations (roughly 5% of the 800 plotted sentences' representation), 88% (37) are the original – noisy – UGC samples of PMUMT. Secondly, multi-VNMT embeddings have higher sparsity (0.190 compared to 0.185 of NF-VNMT), which suggests that our model has more structured representations (Bach et al. 2011), which we link to MDN multiple components in Section 7.7, seemingly contributing to enforce sparsity.

Latent encoding and performance In order to visualize if the latent space is linked to MT performance, we report, in Figure 7.5 the noisy sentences and we represent their character-level edit distance metric comparing predictions to the ground truth using 4 quantiles, with cumulative probability distribution partitions of 25%, 50% and 75%, respectively. On one hand, we first considered the word-level edit distance, which gave us equal sets of bin delimiters for the quantiles $[0.55, 0.64, 0.87]$ for our model and the VNMT baseline. On the other hand, character-level edit distance resulted in multi-VNMT having a better performance distribution (smaller edit distance), with $[0.297, 0.433, 0.563]$ quantiles' delimiters, compared to $[0.304, 0.448, 0.583]$ of the baseline, which also hints that word-level BLEU might not uncover the real translation



(a) NF−VNMT (b) multi−VNMT

Figure 7.4: T-SNE representation of the latent space for noisy and normalized versions of PMUMT sentences during evaluation.

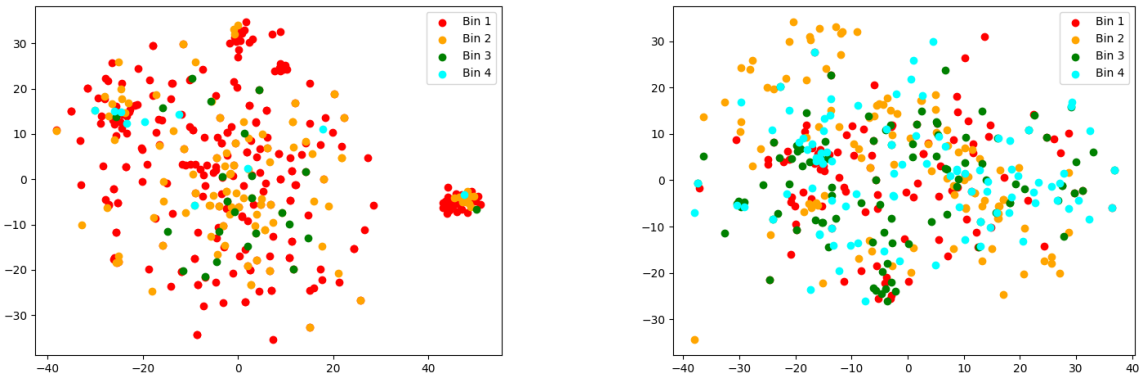


(a) NF−VNMT (b) multi−VNMT

Figure 7.5: T-SNE representation of the latent space for noisy PMUMT sentences during evaluation. Color portrays the quantile of character edit distance between prediction and reference. *Q1* contains the best translations (lowest edit metric).

performance gap in our benchmarking. Figure 7.5 does not show clear clusters of performance, however, multi−VNMT seems to perform worse for vectors with larger norm, whereas performances appear to be more uniformly-distributed for the baseline.

Latent space recovering from noise To measure the perturbations that the model suffers when source-side noise is present, we compare how the latent representations built by our two VNMT models of the noisy PMUMT match those of their corresponding normalized version. In Figure 7.6, we plot the same dimensional reduced latent space and we encode color for their bins of cosine similarity of the hereby shown noisy sentences to their corresponding normalized version. The bins for both plots were chosen using partitions' delimiters $[0.30, 0.44, 0.57]$. This was done to compare both latent spaces with the same similarity values' bins, however, multi−VNMT has



(a) NF−VNMT (b) multi−VNMT

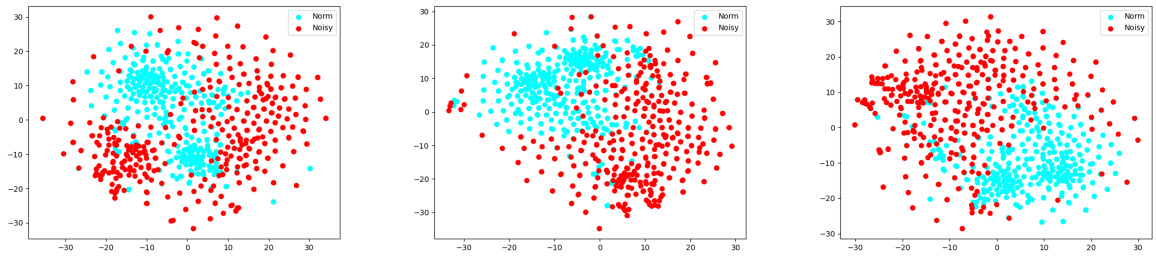
Figure 7.6: T-SNE representation of the latent space for noisy PMUMT sentences during evaluation. Color portrays the quantile of cosine similarity between noisy and normalized versions. *Q1* contains the samples with the least distance (worse performance).

overall higher metric quantiles ($[0.24, 0.36, 0.45]$) compared to NF−VNMT ($[0.19, 0.30, 0.40]$), which suggests that the multi−VNMT latent representations are more robust to UGC.

7.5.2 More robust embeddings for UGC

In this section, we study the source embeddings of our models to assess whether VNMT promotes learning more robust embeddings that could prove valuable for larger-scale transfer learning models. We compare noisy and normalized versions of the FR PMUMT source side to assess whether they have a closer representation.

Noisy vs canonical data We now study the embeddings learned by multi−VNMT and assess how noise affects them compared to those of the baselines. We computed the pair-wise cosine similarity between corresponding PMUMT noisy and normalized samples' source embeddings learned by Transformer Base, NF−VNMT and multi−VNMT, which resulted in 0.706, 0.744 and 0.750, respectively. This quantifies how VNMT can enforce learning more robust source representations since noisy UGC sentences are more related to their normalized version than for the baseline. We display the source embeddings for the three NMT systems in Figure 7.7 and we mark the noisy and normalized corpus's versions in red and blue, respectively. Each observation in the graph corresponds to the embedding of each sentence, computed by taking the average of the token-level embeddings. We can notice how both VNMT systems have a tendency to separate noisy and normalized sentences compared to Transformer Base, and



(a) `Transformer Base` (0.706) (b) `NF-VNMT` (0.744) (c) `multi-VNMT` (0.750)

Figure 7.7: T-SNE representation of the encoder embeddings for noisy and corresponding normalized `PMUMT` sentences during evaluation. *Average cosine similarity between corresponding noisy and normalized version of the `PMUMT` evaluation framework are reported between parentheses for each NMT system.*

having, at the same time, higher cosine similarity. This seems to indicate that VNMT captures some noise structure information, as a separation between noisy and normalized text becomes more evident (Carbonnelle and Vleeschouwer 2021), as opposed to our results on the latent space learning representations, discussed previously.

Recovering from UGC specificities Likewise, we now report the cosine similarity between sentences with only an unique type of UGC specificity, following the isolation and study of individual and controlled UGC phenomena discussed in Chapter 6.

Bootstrapping VNMT embeddings As discussed above, in Figure 7.7 we noticed that VNMT seems to enforce noisy morphology modeling to the Transformer's embeddings in an implicit fashion. This motivated us to study whether the information in such learning representations can be used by the `Transformer Base` backbone model and benefit from improved robustness without the direct latent space contribution, i.e. removing all the VI blocks, thus using only the backbone model. This is also computationally favorable, since the number of parameters of the final model matches the backbone architecture, avoiding all the complexity overhead introduced by the VI modules. Thus, we report BLEU scores for the `Transformer Base` model trained on `OpenSubtitles`, by either initializing the VNMT-pretrained embeddings or fine-tuning (FT) the system. We have performed FT using the same data configuration as in `OpenSubtitles` [5] and continued training for 3 epochs from the `Transformer Base` model in Table 7.1 while replacing the embeddings by their VNMT-learned version's weights.

[5] It is worth highlighting that there is no difference between the corpus used for any of the transfer-learning MT systems and the one used for training the benchmark in Table 7.1.

	PFSMB †	MTNT †	News	OpenSubTest ◇
Transformer Base	27.7	28.4	26.4	31.4
multi-VNMT	28.4	29.2	26.4	31.5
Pretrained init.	**29.0**	28.2	26.2	31.3
Frozen embs.	28.4	**28.9**	**26.8**	31.3
Fine-tuned	28.4	**28.9**	26.5	**31.4**

Table 7.7: VNMT-learned source embeddings to transfer robust representations to the Transformer Base model.

Results in Table 7.7 provide evidence that VNMT enforces more robust embeddings, which perform significantly better over the PFSMB UGC test set compared to the baseline, the system Frozen embs. giving the most consistent results over UGC. This system also keeps good performance over the newstest'14 canonical OOD test set, while taking advantage of an increased robustness to UGC. Such an improvement alleviates the loss of performance over newstest'14 in our previous results, which was the only test set for which multi-VNMT underperformed the non-VNMT baseline in Table 7.1. However, since the embeddings are less prone to over-fitting, the in-domain OpenSubTest's results are -0.1 BLEU points less than Transformer Base. Only the Fine-tuned model was able to maintain the same in-domain performance. These results indicate that VNMT promotes robustness to the NMT backbone and could be useful for achieving more robust pretrained embeddings.

7.6 Blind test sets scores

We now evaluate our best performing model (multi-VNMT trained on OpenSubtitles) on the blind test sets used previously during this dissertation, translating never-seen UGC tests, to assess whether our approach proves valuable for generalization over a larger spectrum of UGC types. We have also included the 4Square corpus (Berard et al. 2019a) to validate our VNMT system on other domain of UGC (restaurant reviews). We also display the results when using the NF-VNMT baseline and the Transformer Base model to assess improvement of our proposed architecture for such test sets. We report such results in Table 7.8, where we can see that multi-VNMT consistently outperforms the baselines for our blind UGC test sets, including the 4Square corpus, which it UGC domain differs from the PFSMB and MTNT social-media

	PFSMB (Blind)	MTNT (Blind)	4Square
Transformer Base	19.7	25.0	21.9
+FT emb.	19.4	25.3	22.0
NF-VNMT	**20.0**	25.3	22.0
multi-VNMT	**20.0**	**26.4**	**22.5**

Table 7.8: BLEU scores of our best systems on UGC blind test sets.

discussion corpora. It is very interesting to notice that, although the in-domain performances for these 3 systems are very similar (between 31.4 and 31.5 BLEU in Table 7.1), the performance gap of blind UGC test sets is considerably larger, roughly +1.7 BLEU. We have also added the best transfer-learning model from the previous section (Transformer Base +FT emb.) and showed an increase of performance when translating MTNT and 4Square blind test sets by +0.3 and +0.1 BLEU, respectively. However, contrary to the VNMT models, we noticed that +FT emb. was outperformed by the baseline, suffering a performance detriment of 0.3 BLEU.

7.7 How do MDN's components react to UGC?

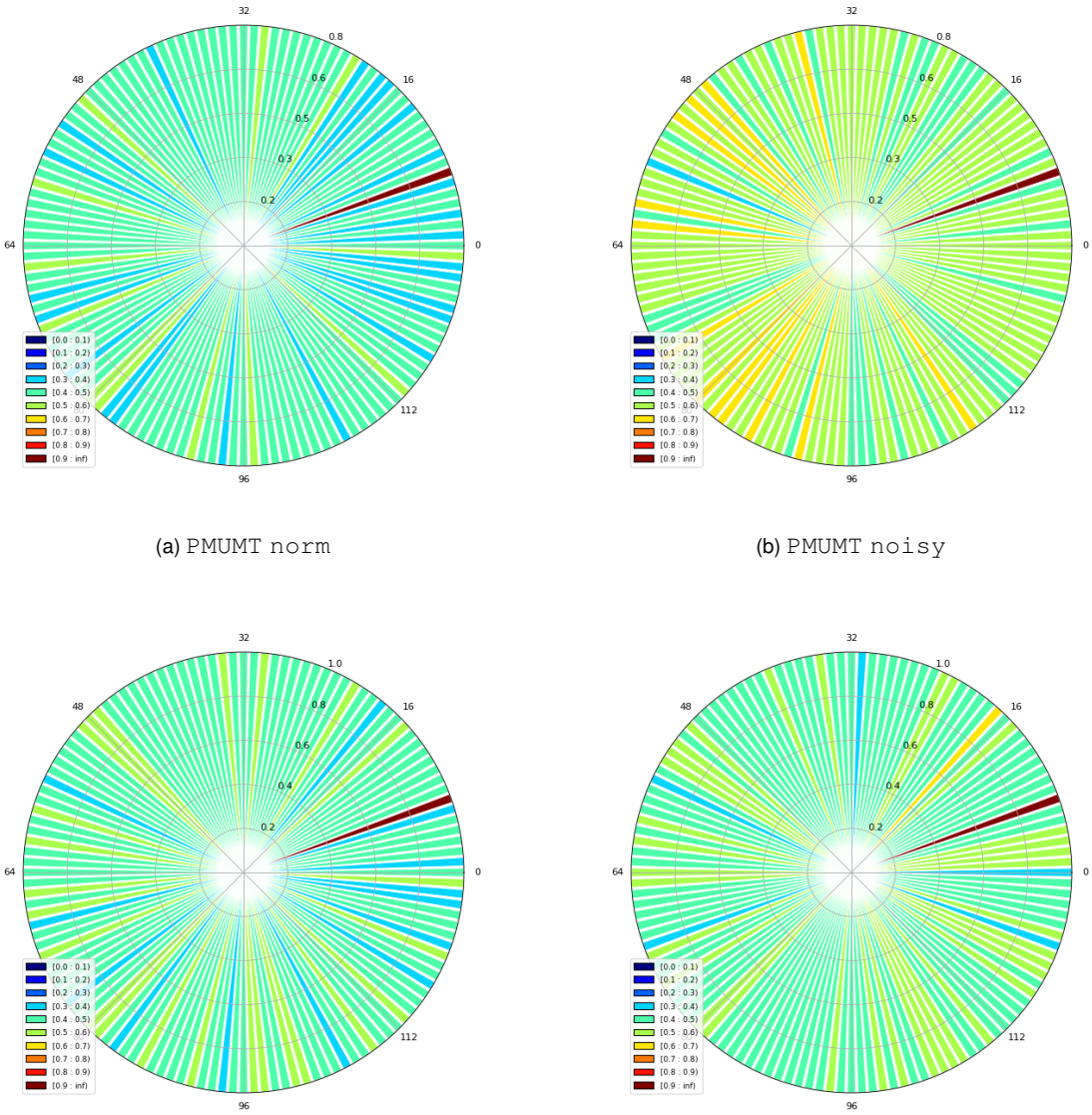
We now proceed to analyze and visualize how the MDN mixture coefficients react when translating our different test sets. In order to do so, in Figure 7.8 we report results for the canonical test sets, the normalized PMUMT corpus, and its noisy original UGC version. Each bar of the Wind Rose diagram represents one of the 128 independent trained distributions' mixture weights, which have been normalized and scaled across the four graphics, and where the 7th MDN component seems to be consistently the one that drives most of the decoding for the presented experiments. Furthermore, we can notice that most mixing coefficients are, for the most part, have around 50% probability of contributing to the final inference mixture, despite not enforcing this behavior with any specific method (e.g. dropout). On the other hand, the visualization suggests that both yellow (50-60%) and blue components (30-40% of activation) are variable across test sets, being very similar between PMUMT Norm and OpenSubTest, which could indicate that the mixture components are learning to encode different types of texts, potentially working as an implicit topic modeling module. Regarding the visualization when translating PMUMT Noisy, the main MDN component identified above, seems less important even when compared to the out-of-domain newstest'14 set, which suggests that the MDN uses more

dense representations when processing noisy texts. Comparing the mixture coefficients in the figure, we can notice that the noisy UGC PMUMT and the out-of-domain newstest'14, diverge from the in-domain OpenSubTest and normalized UGC PMUMT corpus.

7.8 VNMT for other languages: Japanese UGC

Datasets and experiments In order to assess whether our results on UGC translation hold for different languages, we trained the same OpenSubtitles data configuration, i.e. 9.2M sentences with 16K BPE tokenization, using the Ja-En language pair. In addition, we segmented Japanese using Kytea (Neubig et al. 2011) and the Transformer Base with unchanged training hyperparameter is used for both multi-VNMT and baseline. For evaluating these systems, we report results using the JESC (Pryzant et al. 2018) subtitles corpus and KFTT (Neubig et al. 2011) as both in-domain and OOD canonical corpora, respectively, as well as, MTNT's Ja-En UGC corpus.

Discussion of UGC Ja-En MT task In order to assess whether our results on UGC translation hold for different languages, we trained the same OpenSubtitles data configuration using the Ja-En language pair (2.1 M sentences) with 16K BPE tokenization. In addition, we segmented Japanese using Kytea as pre-processing. We train for 120K iterations and use a target word drop rate of 0.4. For evaluating these systems, we report results using the JESC (Pryzant et al. 2018) subtitles corpus as in-domain test set, as well as the MTNT's Ja-En UGC corpus (Michel and Neubig 2018). Results are reported in Table 7.9, which showed that multi-VNMT performs better than the non-latent baseline when translating the MTNT and the in-domain JESC test sets with a +0.4 and +0.3 BLEU improvement respectively. Despite the different experimental conditions, it can be interesting to have an overview of the level of performance to expect in different language conditions. Table 7.9 shows that our results are on par with previously reported scores, even if the settings are not comparable (we have trained a transformer-based model with fewer data compared to Michel and Neubig (2018) and a different architecture than Pryzant et al. (2018), as well.)



(a) PMUMT norm

(b) PMUMT noisy

(c) OpenSubTest

(d) newstest'14

Figure 7.8: Average MDN mixture weights for test sets of different natures.

	MTNT †	JESC◇
Transformer	5.90	11.50
multi-VNMT	**6.30**	**11.80**
(Michel and Neubig 2018)	6.65	—
(Michel and Neubig 2018) (+tuning)	9.82	—
(Pryzant et al. 2018)	—	6.30

Table 7.9: BLEU scores of our translation systems trained on OpenSubtitles Ja-En. The † symbol indicates the UGC test sets, and ◇ in-domain test sets.

7.9 Conclusion and perspectives

Our results prove that we can achieve consistent performance improvement by using extended VNMT architectures that employ MDN, for which, N independent multi-way variational posterior approximations are produced and mixed as in Gaussian Mixture Model. Our approach adds roughly $3.5M$ parameters (+5%) and, systematically, outperforms previous Transformer-based VNMT using normalizing flows on MT, witnessing the largest performance gap for our very noisy `PFSMB` corpus.

In addition, by exploring the learning representations trained by our VNMT model, and through conducting transfer learning experiments with such, we investigate the robustness brought to UGC, and show that VNMT enforces such property to the backbone model, bringing a promising avenue for more robust pretrained neural learning representations. However, an open question arising from this work, it is currently unclear if the performance gain we observed is due to a better generalisation to distributional shift or if it corresponds to a better adaptation to noise in the input. Future works will be devoted to this question, which can be abstracted away to study whether UGC idiosyncrasies are a form of noise, some parts being learnable, or are rather points to a new domain. We report interesting but mitigated results when using an accessory source reconstruction loss to improve robustness, which we plan to study in the future using other sorts of monolingual data and training protocols, such as denoising autoencoders and monolingual source-side UGC corpora.

Chapter 8

Conclusions and perspectives

The goal of this work is to improve the robustness of Neural Machine Translation (NMT) when processing noisy social media user-generated content (UGC). Such texts pose challenges due to their extensive forms of expression and multi-domain nature, as explained in Chapter 3. We have focused on the zero-shot translation scenario, in which the training data is restricted to publicly available canonical corpora, and developed methods that are agnostic to the target distribution of UGC, and can better generalize over a wide range of UGC sources.

Identification and formulation We started by describing NMT and phrase-based statistical machine translation baselines in Chapter 4 and elaborated on the differences in performance on test sets of different nature. Particularly, we focused on the type of errors observed when using NMT to translate UGC and assessed how robust our baselines are. In this regard, neural-based translation methods provided overall best results for canonical test data, while being relatively brittle to UGC noise than PB-SMT.

We therefore focused our research (mainly on the French-English language pair) on NMT, which turned out to be better at translating in-domain and out-domain canonical corpora, but not robust to UGC compared to PB-SMT. An analysis of the attention mechanism provides a first explanation for NMT's inadequacy to translate UGC: NMT indeed predicts translations with a less similar length distribution than the references, caused either by omission of translations, over-translation due to failure of the attention mechanism, or hallucinations.

Using pronunciation to correct from noise As a first attempt to recover from UGC specificities, in Chapter 5 we proposed a normalization pipeline to correct phonetic writing (discussed in Section 5.1) by finding potential normalized candidates using pronunciation similarities. A ranking of such candidates by a language model (LM) provides the most likely corrected sentence. As discussed, our results showed interesting normalization cases and slightly improved results over UGC translation, without requiring any supplementary data. However, they showed limited impact on blind UGC test sets and suffered from artificial noise by confusing potentially normalized tokens. Regarding this last aspect, the identical French pronunciation of certain plural forms was shown to introduce erroneous correction of tokens that otherwise should not have been changed. Relative to the reported caveats of our methods, the study of an end-to-end trained NMT phonetic embeddings, providing both original and pronunciation information to output the best translation.

Understanding UGC specificities and fine granularity impact on MT In order to account for a wider range of UGC phenomena, we explored the use of character-based representations to assess their robustness to UGC in Chapter 6. We have highlighted the importance of the vocabulary size for char-based systems, a parameter whose importance had never been discussed until now.

These experiments also motivated the creation of a novel UGC evaluation framework with a fine-grained typology and the possibility to isolate different specificities of UGC, a necessary step to understand the mechanism at play while translating noisy UGC. This framework aims to evaluate how different models behave under different kinds of noise. Our results showed that character-level NMT is less sensitive to misspellings (in the form of letter addition or deletion, incorrect diacritics, and incorrect verb tense), as well as incorrect tokenization and inconsistent casing. However, character-based systems were generally outperformed by subword segmentation, which performed better in our experimental setup.

Improving robustness to UGC via latent variable methods In the last part of this dissertation (Chapter 7), we explored Variational Neural Machine Translation (VNMT) and presented a novel architecture using Mixture Density Networks to perform multimodal Variational Inference. We introduce mixture models to perform VI in a multimodal information flow, which showed

improved robustness capabilities when translating UGC. Through probing and visualization methods, we investigated how latent codes can be useful to recover from noise by enforcing robust learning representations, while mitigating overfitting and thus improving out-of-domain translation performance. We conducted several experiments to evaluate how the representations of the VNMT models are less sensitive to UGC-related perturbations compared to a non-latent baseline. Our proposed approach showed improved robustness capabilities to UGC, without hurting translation quality on canonical (in-domain and out-domain) test sets.

Six years after the ParSiTi project proposal: progress on UGC translation In recent years, much of the attention in NLP has been focused on Large Language Models (LLM), which have been shown to be able to capture high-order information structures and exhibit interesting properties in zero-shot scenarios (Kojima et al. 2022). Surprisingly, these capabilities arise from a relatively simple language modeling (LM) task, i.e. masked LM, which allows the exploitation of massive amounts of text in an unsupervised manner. This property, combined with the possibility of instant access to huge and extremely diverse corpora thanks to the Internet, has led to the development of text-based systems with impressive generative power and flexibility (Guardian 2022). This approach makes it possible to exploit UGC data without the need of producing reference translations, thus enhancing NMT with an apparent advantage when translating UGC. To illustrate the case, in Figure 8.1, we revisit the original UGC sample studied in Chapter 1 translated by popular translation engines, some using LLM.

In Figure 8.1 we can observe that UGC translation is still much of a work in progress to this day: in most cases, the French UGC-specific tokens (e.g. "*tabuz*", correctly spelled "*t'abuses*") are left unchanged in their English translation, apparently due to the MT systems treating them as Named Entities.

It can, however, be noticed that GPT-based (Brown et al. 2020) systems succeed in translating two noisy UGC tokens: BingGPT© correctly translates "*Ki ca ?*" into "*Who is it ?*" and "*lavé rien fé*" into "*did nothing*", correcting French phonetic writing on both; ChatGPT© is able of partially translating "*lemecousonchien?*" into "*The guy with the dog?*", recovering from the missing word segmentation, even if it mistranslates "ou" into "*with*" rather than "*or*".

These examples show that, despite relying on large amounts of text, most of them gathered from the Internet, state-of-the-art MT engines have made little progress in UGC translation,

(@rigolboche)

Original source
T'as vu il l'a bien cherché wsh #AperoChezRicard → +10000, shah! → tabuz, lavé rien fé → ki ca ? lemecousonchien ? → Wtf is wrong with him ? #PETA4EVER → ki ca ? le chien ? looool

Bing© (2016)	Bing© (2023)
Did you see he looked for it wsh #AperoChezRicard → +10000, shah! → tabuz, washed nothing fe → ki ca ? theguyorhisdog? → Wtf is wrong with him ? #PETA4EVER. → ki ca ? the dog? looool	Did you see he looked for it wsh #AperoChezRicard → +10000, sheh! → tabuz, washed nothing fe → Kica? Lemecousonchien? → Wtf is wrong with him ? #PETA4EVER. → ki ca? The dog? looool
Google Translator© (2023)	**DeepL© (2023)**
Did you see he looked for it wsh #AperoChezRicard → +10000, sheh! → tabuz, washed nothing fe → Kica? Lemecousonchien? → Wtf is wrong with him ? #PETA4EVER → ki ca? The dog? looool	You see he had it coming wsh #AperoChezRicard → +10000, shah! → tabuz, washed nothing fe → kica? lemecousonchien? → Wtf is wrong with him ? #PETA4EVER. → ki ca? the dog? looool
BingGPT© (2023)	**ChatGPT© (2023)**
Did you see he deserved it wsh #AperoChezRicard → +10000, sheh! → tabuz, did nothing → kica? lemecousonchien? → Wtf is wrong with him ? #PETA4EVER → who is it ? the dog ? looool	Did you see he was asking for it, wsh #AperoChezRicard → +10000, sheh! → Tabuz, didn't wash anything → Kica? The guy with the dog? → What's wrong with him? #PETA4EVER. → Ki ca? The dog? Looool.

Figure 8.1: Typical social media thread initiated by a seed photo and its automatic translation. *Inspired from a real conversation about a series of demonstrations that took place in Greece. Bing was used (in April 2016) as it was then the official MT engine for Twitter and Facebook.*

six years after the ParSiTi project started. This observation highlights the inadequacies of utilizing large amounts of training data to "understand" UGC, a never productive domain, and, consequently, justify the zero-shot setting we have adopted in this work.

General conclusions

In this work, we have addressed the automatic translation of UGC and the challenges it poses, both in terms of target distribution drift and robustness to noise. Indeed, a very rich productivity lies behind the amazing nature of UGC: a rich hub for emergent communication derived from user interactions. This observation motivates our focus on the zero-shot scenario, i.e., we intended to study and improve the generalization capabilities of the models without any or minimal effort to try to match the target distribution, nor any attempt to tailor the data source to the task throughout this dissertation.

Regarding our efforts to improve UGC MT performance, we started our work by producing a lexical normalizer based on specific heuristics (i.e., aiming to correct phonetic spelling), in order to later broaden our scope and analyze a larger set of UGC specificities, thus directing our research towards improving and evaluating the robustness of NMT for UGC. To this end, we found our proposed UGC evaluation framework helpful, allowing us to both compare different NMT architectures and assess the impact of well-defined UGC phenomena on performance. Regarding our research on NMT, popular character-level NMT architectures showed interesting robustness properties, being relatively more robust than their subword BPE counterparts, but lacking overall in-domain performance in our experiments.

In the final stage of this research, we obtained promising results when exploring Transformer-based VNMTs, which provide consistently better results than their non-latent backbones and improved robustness to UGC. As part of our experiments, we proposed a VNMT model with Mixture Density Networks, which results in better translation quality, especially when when processing UGC. In addition, by exploring these models, we show evidence of VNMT's robustness capabilities to UGC and, Perhaps more interestingly, we report experiments and visualizations suggesting that VNMT forces more robust learning representations on the backbone model, which can later be exploited without the VI blocks and retained after transfer (e.g., for fine-tuning).

In summary, our research has demonstrated the complex and extremely variable nature of UGC, requiring us to expect the improbable during evaluation, motivating our studies on generalization and robustness. As a complementary framework, we developed evaluation protocols and resources to characterize the impact of UGC on performance, making our best efforts to avoid reducing UGC specificities to simple *noise*. Finally, variational architectures

were proposed as a promising avenue for UGC translation, which, compared to other domain adaptation methods such as domain specialization (i.e., fine-tuning), is data-agnostic and maintains optimal performance for in-domain test sets.

Bibliography

Hassan Al-Haj and Alon Lavie. The impact of arabic morphological segmentation on broad-coverage english-to-arabic statistical machine translation. *Machine Translation*, 26(1/2):3–24, 2012. ISSN 09226567, 15730573.

Rami Al-Rfou, Dokook Choe, Noah Constant, Mandy Guo, and Llion Jones. Character-level language modeling with deeper self-attention. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 3159–3166. AAAI Press, 2019. doi: 10.1609/aaai.v33i01.33013159.

Cyril Allauzen, Michael Riley, Johan Schalkwyk, Wojciech Skut, and Mehryar Mohri. Openfst: A general and efficient weighted finite-state transducer library. In *Proceedings of the 12th International Conference on Implementation and Application of Automata*, CIAA'07, pages 11–23, Berlin, Heidelberg, 2007. Springer-Verlag. ISBN 3-540-76335-X, 978-3-540-76335-2.

A. Anastasopoulos. An analysis of source-side grammatical errors in NMT. *CoRR*, 2019.

Ali Araabi, Christof Monz, and Vlad Niculae. How effective is byte pair encoding for out-of-vocabulary words in neural machine translation? *CoRR*, abs/2208.05225, 2022. doi: 10.48550/arXiv.2208.05225.

Duygu Ataman and Marcello Federico. An evaluation of two vocabulary reduction methods for neural machine translation. In Colin Cherry and Graham Neubig, editors, *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas, AMTA 2018, Boston, MA, USA, March 17-21, 2018 - Volume 1: Research Papers*, pages 97–110. Association for Machine Translation in the Americas, 2018.

Duygu Ataman, Orhan Firat, Mattia Antonino Di Gangi, Marcello Federico, and Alexandra Birch. On the importance of word boundaries in character-level neural machine translation. In Alexandra Birch, Andrew M. Finch, Hiroaki Hayashi, Ioannis Konstas, Thang Luong, Graham Neubig, Yusuke Oda, and Katsuhito Sudoh, editors, *Proceedings of the 3rd Workshop on Neural Generation and Translation@EMNLP-IJCNLP 2019, Hong Kong, November 4, 2019*, pages 187–193. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-5619.

Francis R. Bach, Rodolphe Jenatton, Julien Mairal, and Guillaume Obozinski. Structured sparsity through convex optimization. *CoRR*, abs/1109.2397, 2011.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015.

Azlin Zanariah Bahtar and Mazzini Muda. The impact of user – generated content (ugc) on product reviews towards online purchasing – a conceptual framework. *Procedia Economics and Finance*, 37:337–342, 2016. ISSN 2212-5671. doi: https://doi.org/10.1016/S2212-5671(16)30134-4. The Fifth International Conference on Marketing and Retailing (5th INCOMaR) 2015.

David Bamman. Natural language processing for the long tail. In Rhian Lewis, Cecily Raynor, Dominic Forest, Michael Sinatra, and Stéfan Sinclair, editors, *12th Annual International Conference of the Alliance of Digital Humanities Organizations, DH 2017, Montréal, Canada, August 8-11, 2017, Conference Abstracts*. Alliance of Digital Humanities Organizations (ADHO), 2017.

Marion Baranes. *Spelling Normalisation of Noisy Text*. Theses, Université Paris-Diderot - Paris VII, October 2015.

Yonatan Belinkov and Yonatan Bisk. Synthetic and natural noise both break neural machine translation. In *Proceedings of the 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.

Luisa Bentivogli, Arianna Bisazza, Mauro Cettolo, and Marcello Federico. Neural versus phrase-based machine translation quality: a case study. In *EMNLP*, 2016.

Alexandre Berard, Ioan Calapodescu, Marc Dymetman, Claude Roux, Jean-Luc Meunier, and Vassilina Nikoulina. Machine translation of restaurant reviews: New corpus for domain adapta-

tion and robustness. In Alexandra Birch, Andrew M. Finch, Hiroaki Hayashi, Ioannis Konstas, Thang Luong, Graham Neubig, Yusuke Oda, and Katsuhito Sudoh, editors, *Proceedings of the 3rd Workshop on Neural Generation and Translation@EMNLP-IJCNLP 2019, Hong Kong, November 4, 2019*, pages 168–176. Association for Computational Linguistics, 2019a. doi: 10.18653/v1/D19-5617.

Alexandre Berard, Ioan Calapodescu, and Claude Roux. Naver labs europe's systems for the WMT19 machine translation robustness task. In Ondrej Bojar, Rajen Chatterjee, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno-Yepes, Philipp Koehn, André Martins, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana L. Neves, Matt Post, Marco Turchi, and Karin Verspoor, editors, *Proceedings of the Fourth Conference on Machine Translation, WMT 2019, Florence, Italy, August 1-2, 2019 - Volume 2: Shared Task Papers, Day 1*, pages 526–532. Association for Computational Linguistics, 2019b. doi: 10.18653/v1/w19-5361.

Alexandre Berard, Calapodescu Ioan, and Claude Roux. NAVER LABS Europe's Systems for the WMT19 Machine Translation Robustness Task. In *Proceedings of the Fourth Conference on Machine Translation, Volume 2: Shared Task Papers*, Florence, Italy, August 2019c. Association for Computational Linguistics.

Nicola Bertoldi, Barry Haddow, and Jean-Baptiste Fouet. Improved minimum error rate training in moses. *Prague Bull. Math. Linguistics*, 91:7–16, 2009.

Christopher M. Bishop. Mixture density networks. Technical report, 1994.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *TACL*, 5:135–146, 2017.

Ondrej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno-Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana L. Neves, Martin Popel, and Matt Post et al. Findings of the 2016 conference on machine translation. In *WMT*, 2016.

Alessandro Bondielli and Francesco Marcelloni. A survey on fake news and rumour detection techniques. *Inf. Sci.*, 497:38–55, 2019. doi: 10.1016/j.ins.2019.05.035.

Keith Broni. Emoji use at all-time high. https://blog.emojipedia.org/emoji-use-at-all-time-high/, 2021. Accessed: 2021-08-16.

P. Brown, J. Cocke, S. Della Pietra, V. Della Pietra, F. Jelinek, R. Mercer, and P. Roossin. A statistical approach to language translation. In *Proceedings of the 12th Conference on Computational Linguistics - Volume 1*, COLING '88, page 71–76, USA, 1988. Association for Computational Linguistics. ISBN 963 8431 56 3. doi: 10.3115/991635.991651.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Rafael Calvo, David Milne, Sazzad Hussain, and Helen Christensen. Natural language processing in mental health applications using non-clinical texts. *Natural Language Engineering*, 23: 1–37, 01 2017. doi: 10.1017/S1351324916000383.

Simon Carbonnelle and Christophe De Vleeschouwer. Intraclass clustering: an implicit learning ability that regularizes dnns. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

S. Castilho, J. Moorkens, F. Gaspari, R. Sennrich, V. Sosoni, Y. Georgakopoulou, P. Lohar, A. Way, A. Valerio, Miceli B., and M. Gialama. A comparative quality evaluation of pbsmt and nmt using professional translators. 2017.

Nathanael Chambers, Timothy Forman, Catherine Griswold, Kevin Lu, Yogaish Khastgir, and Stephen Steckler. Character-based models for adversarial phone extraction: Preventing human sex trafficking. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-*

NUT 2019), pages 48–56, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-5507.

Yucheng Chen. Twitter- a new pathway to access product innovation ideas - can machine learning help pepsico identify innovative ideas in user-generated content platforms?, June 2022.

Colin Cherry, George Foster, Ankur Bapna, Orhan Firat, and Wolfgang Macherey. Revisiting character-based neural machine translation with capacity and compression. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 4295–4305, 2018.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1724–1734. ACL, 2014. doi: 10.3115/v1/d14-1179.

Arindam Chowdhury and Lovekesh Vig. An efficient end-to-end neural model for handwritten text recognition. In *British Machine Vision Conference 2018, BMVC 2018, Newcastle, UK, September 3-6, 2018*, page 202. BMVA Press, 2018.

Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555, 2014.

Marta R. Costa-jussà and José A. R. Fonollosa. Character-based neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 2: Short Papers*, 2016.

Oxford Language Dictionary. Emoticons, 2017.

Shuoyang Ding, Adithya Renduchintala, and Kevin Duh. A call for prudent choice of subword merge operations in neural machine translation. In Mikel L. Forcada, Andy Way, Barry Haddow, and Rico Sennrich, editors, *Proceedings of Machine Translation Summit XVII Volume*

1: Research Track, MTSummit 2019, Dublin, Ireland, August 19-23, 2019, pages 204–213. European Association for Machine Translation, 2019.

Yanzhuo Ding, Yang Liu, Huanbo Luan, and Maosong Sun. Visualizing and understanding neural machine translation. In Regina Barzilay and Min-Yen Kan, editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1150–1159. Association for Computational Linguistics, 2017. doi: 10.18653/v1/P17-1106.

M. Dowling, T. Lynn, A. Poncelas, and A. Way. SMT versus NMT: preliminary comparisons for irish. In *LoResMT@AMTA*, 2018.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia, June 2013. Association for Computational Linguistics.

Javid Ebrahimi, Daniel Lowd, and Dejing Dou. On adversarial examples for character-level neural machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 653–663, 2018.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1045.

Jacob Eisenstein. What to do about bad language on the internet. In *Proceedings of the 2013 conference of the North American Chapter of the association for computational linguistics: Human language technologies*, pages 359–369, 2013.

Jennifer Foster. “cba to check the spelling”: Investigating parser performance on discussion forum posts. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 381–384, Los Angeles, California, June 2010. Association for Computational Linguistics.

Ryo Fujii, Masato Mita, Kaori Abe, Kazuaki Hanawa, Makoto Morishita, Jun Suzuki, and Kentaro Inui. Phemt: A phenomenon-wise dataset for machine translation robustness on user-generated contents. In Donia Scott, Núria Bel, and Chengqing Zong, editors, *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 5929–5943. International Committee on Computational Linguistics, 2020. doi: 10.18653/v1/2020.coling-main.521.

Mercedes Garcia-Martinez, Loïc Barrault, and Fethi Bougares. Factored Neural Machine Translation Architectures. In *International Workshop on Spoken Language Translation (IWSLT'16)*, Seattle, United States, 2016.

Carlisle E. George and Jackie Scerri. Jackie, web 2.0 and user-generated content: Legal challenges in the new frontier. *Journal of Information, Law and Technology*, 2, 2007.

Raul Gomez, Jaume Gibert, Lluís Gómez, and Dimosthenis Karatzas. Exploring hate speech detection in multimodal publications. In *IEEE Winter Conference on Applications of Computer Vision, WACV 2020, Snowmass Village, CO, USA, March 1-5, 2020*, pages 1459–1467. IEEE, 2020. doi: 10.1109/WACV45572.2020.9093414.

Alex Graves, Santiago Fernández, and Jürgen Schmidhuber. Bidirectional LSTM networks for improved phoneme classification and recognition. In Wlodzislaw Duch, Janusz Kacprzyk, Erkki Oja, and Slawomir Zadrozny, editors, *Artificial Neural Networks: Formal Models and Their Applications - ICANN 2005, 15th International Conference, Warsaw, Poland, September 11-15, 2005, Proceedings, Part II*, volume 3697 of *Lecture Notes in Computer Science*, pages 799–804. Springer, 2005. doi: 10.1007/11550907_126.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1640, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1154.

The Guardian. Facebook translates 'good morning' into 'attack them', leading to arrest, 2017. URL https://www.theguardian.com/technology/2017/oct/24/facebook-palestine-israel-translates-good-morning-attack-them-arrest.

The Guardian. Ai bot chatgpt stuns academics with essay-writing skills and usability, 2022. URL https://www.theguardian.com/technology/2022/dec/04/ai-bot-chatgpt-stuns-academics-with-essay-writing-skills-and-usability.

Emil Julius Gumbel. *Statistical theory of extreme values and some practical applications; a series of lectures*. Applied mathematics series ; 33. U.S. Govt. Print. Office, Washington, 1954.

David Ha and Douglas Eck. A neural representation of sketch drawings. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 2455–2467, 2018.

Hesam Haddad, Hakimeh Fadaei, and Heshaam Faili. Handling oov words in nmt using unsupervised bilingual embedding. In *2018 9th International Symposium on Telecommunications (IST)*, pages 569–574, 2018. doi: 10.1109/ISTEL.2018.8661016.

Barry Haddow and Philipp Koehn. Analysing the effect of out-of-domain data on SMT systems. In Chris Callison-Burch, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia, editors, *Proceedings of the Seventh Workshop on Statistical Machine Translation, WMT@NAACL-HLT 2012, June 7-8, 2012, Montréal, Canada*, pages 422–432. The Association for Computer Linguistics, 2012.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8): 1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

S. Jean, K. Cho, R. Memisevic, and Y. Bengio. On using very large target vocabulary for neural machine translation. In *ACL*, 2015.

Valentin Jijkoun, Mahboob Khalid, Maarten Marx, and Maarten Rijke. Named entity normalization in user generated content. pages 23–30, 01 2008. doi: 10.1145/1390749.1390755.

Dan Jurafsky and James H. Martin. *Speech and language processing : an introduction to natural language processing, computational linguistics, and speech recognition*. Pearson Prentice Hall, Upper Saddle River, N.J., 2009. ISBN 9780131873216 0131873210.

Rasoul Kaljahi, Jennifer Foster, Johann Roturier, Corentin Ribeyre, Teresa Lynn, and Joseph Le Roux. Foreebank: Syntactic analysis of customer support forums. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1341–1347, 2015.

Zadeh Kaljahi and Rasoul Samad. *The role of syntax and semantics in machine translation and quality estimation of machine-translated user-generated content*. PhD thesis, Dublin City University, 2015.

Huda Khayrallah and Philipp Koehn. On the impact of various types of noise on neural machine translation. In Alexandra Birch, Andrew M. Finch, Minh-Thang Luong, Graham Neubig, and Yusuke Oda, editors, *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation, NMT@ACL 2018, Melbourne, Australia, July 20, 2018*, pages 74–83. Association for Computational Linguistics, 2018. doi: 10.18653/v1/w18-2709.

Giti Khoshamooz and Mohammad Taleai. Multi-domain user-generated content based model to enrich road network data for multi-criteria route planning. *Geographical Analysis*, 49(3): 239–267, 2017. doi: https://doi.org/10.1111/gean.12124.

Yoon Kim, Yacine Jernite, David A. Sontag, and Alexander M. Rush. Character-aware neural language models. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 2741–2749, 2016.

Yoon Kim, Sam Wiseman, and Alexander M. Rush. A tutorial on deep latent variable models of natural language. *CoRR*, abs/1812.06834, 2018.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

Guillaume Klein, Yoon Kim, Yuntian Deng, Vincent Nguyen, Jean Senellart, and Alexander M. Rush. Opennmt: Neural machine translation toolkit. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas, AMTA 2018, Boston, MA, USA, March 17-21, 2018 - Volume 1: Research Papers*, pages 177–184, 2018.

P. Koehn and R. Knowles. Six challenges for neural machine translation. In *NMT@ACL*, 2017.

Philipp Koehn. Statistical significance tests for machine translation evaluation. pages 388–395, 01 2004.

Philipp Koehn. *Phrase-Based Models*, page 127–154. Cambridge University Press, 2009. doi: 10.1017/CBO9780511815829.006.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open source toolkit for statistical machine translation. In *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic*, 2007a.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open source toolkit for statistical machine translation. In *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic*, 2007b.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *CoRR*, abs/2205.11916, 2022. doi: 10.48550/arXiv.2205.11916.

John Krumm, Nigel Davies, and Chandra Narayanaswami. User-generated content. *Pervasive Computing, IEEE*, 7:10 – 11, 01 2009. doi: 10.1109/MPRV.2008.85.

Viet Bac Le, Sopheap Seng, Laurent Besacier, and Brigitte Bigi. Word/sub-word lattices decomposition and combination for speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2008, March 30 - April 4, 2008, Caesars Palace, Las Vegas, Nevada, USA*, pages 4321–4324, 2008. doi: 10.1109/ICASSP.2008.4518611.

Jason Lee, Kyunghyun Cho, and Thomas Hofmann. Fully character-level neural machine translation without explicit segmentation. *TACL*, 5:365–378, 2017.

Maoxi Li, Mingwen Wang, Hanxi Li, and Fan Xu. Modeling monolingual character alignment for automatic evaluation of chinese translation. *ACM Trans. Asian Low Resour. Lang. Inf. Process.*, 15(3):16:1–16:18, 2016. doi: 10.1145/2815619.

Zachary Chase Lipton. A critical review of recurrent neural networks for sequence learning. *CoRR*, abs/1506.00019, 2015.

P. Lison, J. Tiedemann, and M. Kouylekov. Opensubtitles2018: Statistical rescoring of sentence alignments in large, noisy parallel corpora. In *LREC*, 2018.

Ding Liu and Daniel Gildea. Syntactic features for evaluation of machine translation. In Jade Goldstein, Alon Lavie, Chin-Yew Lin, and Clare R. Voss, editors, *Proceedings of the Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization@ACL 2005, Ann Arbor, Michigan, USA, June 29, 2005*, pages 25–32. Association for Computational Linguistics, 2005.

Ramon Lobato, Julian Thomas, and Dan Hunter. Histories of user-generated content: Between formal and informal media economies. *International Journal of Communication*, 5(0), 2011. ISSN 1932-8036.

Pintu Lohar. *Machine translation of user-generated content*. PhD thesis, Dublin City University, 2020.

Minh-Thang Luong and Christopher D. Manning. Achieving open vocabulary neural machine translation with hybrid word-character models. In *Proceedings of the 54th Annual Meeting*

of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers. The Association for Computer Linguistics, 2016. doi: 10.18653/v1/p16-1100.

Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In *EMNLP*, 2015.

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

Benjamin Marie, Atsushi Fujita, and Raphael Rubino. Scientific credibility of machine translation research: A meta-evaluation of 769 papers. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 7297–7306. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.acl-long.566.

Héctor Martínez Alonso, Djamé Seddah, and Benoît Sagot. From noisy questions to Minecraft texts: Annotation challenges in extreme syntax scenario. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pages 13–23, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee.

Yuval Marton, Nizar Habash, and Owen Rambow. Improving Arabic dependency parsing with lexical and inflectional morphological features. In *Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 13–21, Los Angeles, CA, USA, June 2010. Association for Computational Linguistics.

Michael McCloskey and Neal J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. volume 24 of *Psychology of Learning and Motivation*, pages 109–165. Academic Press, 1989. doi: https://doi.org/10.1016/S0079-7421(08)60536-8.

Sara Meftah. *Neural Transfer Learning for Domain Adaptation in Natural Language Processing*. Theses, Université Paris-Saclay, March 2021.

Haitao Mi, Baskaran Sankaran, Zhiguo Wang, and Abe Ittycheriah. Coverage embedding models for neural machine translation. In *EMNLP*, 2016.

Paul Michel and Graham Neubig. MTNT: A testbed for machine translation of noisy text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 543–553, 2018.

Marie-Francine Moens, Juanzi Li, and Tat-Seng Chua. *Mining User Generated Content*. Chapman and Hall/CRC, 2014. ISBN 1466557400.

Behrang Mohit, Alla Rozovskaya, Nizar Habash, Wajdi Zaghouani, and Ossama Obeid. The first QALB shared task on automatic text correction for arabic. In *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing, ANLP@EMNLP 20104, Doha, Qatar, October 25, 2014*, pages 39–47, 2014.

Mehryar Mohri. Semiring frameworks and algorithms for shortest-distance problems. *Journal of Automata, Languages and Combinatorics*, 7(3):321–350, January 2002. ISSN 1430-189X.

David R. Mortensen, Patrick Littell, Akash Bharadwaj, Kartik Goyal, Chris Dyer, and Lori S. Levin. Panphon: A resource for mapping IPA segments to articulatory feature vectors. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3475–3484. ACL, 2016.

Zewdie Mossie. Social media dark side content detection using transfer learning emphasis on hate and conflict. In Amal El Fallah Seghrouchni, Gita Sukthankar, Tie-Yan Liu, and Maarten van Steen, editors, *Companion of The 2020 Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, pages 259–263. ACM / IW3C2, 2020. doi: 10.1145/3366424.3382084.

Jonathan Mutal, Lise Volkart, Pierrette Bouillon, Sabrina Girletti, and Paula Estrella. Differences between SMT and NMT output - a translators' point of view. In *Proceedings of the Human-Informed Translation and Interpreting Technology Workshop (HiT-IT 2019)*, pages 75–81, Varna, Bulgaria, September 2019. Incoma Ltd., Shoumen, Bulgaria. doi: 10.26615/issn.2683-0078.2019_009.

Graham Neubig, Yosuke Nakata, and Shinsuke Mori. Pointwise prediction for robust, adaptable japanese morphological analysis. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA - Short Papers*, pages 529–533. The Association for Computer Linguistics, 2011.

Hermann Ney, Ute Essen, and Reinhard Kneser. On structuring probabilistic dependences in stochastic language modelling. *Computer Speech & Language*, 8(1):1–38, 1994.

Xing Niu, Prashant Mathur, Georgiana Dinu, and Yaser Al-Onaizan. Evaluating robustness to input perturbations for neural machine translation. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8538–8544. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.755.

José Carlos Rosales Núñez, Djamé Seddah, and Guillaume Wisniewski. Phonetic normalization for machine translation of user generated content. In Wei Xu, Alan Ritter, Tim Baldwin, and Afshin Rahimi, editors, *Proceedings of the 5th Workshop on Noisy User-generated Text, W-NUT@EMNLP 2019, Hong Kong, China, November 4, 2019*, pages 407–416. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-5553.

Christopher Olah. Understanding LSTM Networks. https://colah.github.io/posts/2015-08-Understanding-LSTMs/, 2015. [Online; accessed 02-July-2020].

Yonatan Oren, Shiori Sagawa, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust language modeling. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 4226–4236. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1432.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A fast, extensible toolkit for sequence modeling. In Waleed Ammar, Annie Louis, and Nasrin Mostafazadeh, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language*

Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Demonstrations, pages 48–53. Association for Computational Linguistics, 2019. doi: 10.18653/v1/n19-4009.

Matthew O'Hern and Lynn Kahle. The empowered customer: User-generated content and the future of marketing. *Global Economics and Management Review*, 18:22–30, 04 2013. doi: 10.1016/S2340-1540(13)70004-5.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318, 2002.

Peyman Passban, Puneeth S. M. Saladi, and Qun Liu. Revisiting robust neural machine translation: A transformer case study. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, pages 3831–3840. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.findings-emnlp.323.

Nicolas Pécheux, Guillaume Wisniewski, and François Yvon. Reassessing the value of resources for cross-lingual transfer of pos tagging models. *Language Resources and Evaluation*, pages 1–34, 2016. ISSN 1574-0218. doi: 10.1007/s10579-016-9362-7.

Maja Popović. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/W15-3049.

Maja Popovic. chrf++: words helping character n-grams. In Ondrej Bojar, Christian Buck, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno-Yepes, Philipp Koehn, and Julia Kreutzer, editors, *Proceedings of the Second Conference on Machine Translation, WMT 2017, Copenhagen, Denmark, September 7-8, 2017*, pages 612–618. Association for Computational Linguistics, 2017. doi: 10.18653/v1/w17-4770.

Matt Post. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference*

on Machine Translation: Research Papers, WMT 2018, Belgium, Brussels, October 31 - November 1, 2018, pages 186–191, 2018.

Victor Prokhorov, Ehsan Shareghi, Yingzhen Li, Mohammad Taher Pilehvar, and Nigel Collier. On the importance of the Kullback-Leibler divergence term in variational autoencoders for text generation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 118–127, Hong Kong, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-5612.

Reid Pryzant, Youngjoo Chung, Dan Jurafsky, and Denny Britz. JESC: japanese-english subtitle corpus. In Nicoletta Calzolari, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Kôiti Hasida, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asunción Moreno, Jan Odijk, Stelios Piperidis, and Takenobu Tokunaga, editors, *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018*. European Language Resources Association (ELRA), 2018.

Mark Przybocki, Gregory Sanders, and Audrey Le. Edit distance: A metric for machine translation evaluation. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy, May 2006. European Language Resources Association (ELRA).

Michael Przystupa. *Investigating the impact of normalizing flows on latent variable machine translation*. PhD thesis, University of British Columbia, 2020.

Long Qin, Ming Sun, and Alexander I. Rudnicky. System combination for out-of-vocabulary word detection. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2012, Kyoto, Japan, March 25-30, 2012*, pages 4817–4820, 2012. doi: 10.1109/ICASSP.2012.6288997.

Akshai Ramesh, Venkatesh Balavadhani Parthasarathy, Rejwanul Haque, and Andy Way. An error-based investigation of statistical and neural machine translation performance on hindi-to-tamil and english-to-tamil. In Toshiaki Nakazawa, Hideki Nakayama, Chenchen Ding, Raj Dabre, Anoop Kunchukuttan, Win Pa Pa, Ondrej Bojar, Shantipriya Parida, Isao Goto, Hidaya Mino, Hiroshi Manabe, Katsuhito Sudoh, Sadao Kurohashi, and Pushpak Bhattacharyya, editors, *Proceedings of the 7th Workshop on Asian Translation, WAT@AACL/IJCNLP 2020,*

Suzhou, China, December 4, 2020, pages 178–188. Association for Computational Linguistics, 2020.

Alan Ramponi and Barbara Plank. Neural unsupervised domain adaptation in NLP—A survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6838–6855, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.603.

Vikas Raunak, Arul Menezes, and Marcin Junczys-Dowmunt. The curious case of hallucinations in neural machine translation. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tür, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 1172–1183. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.naacl-main.92.

Limor Raviv, Antje Meyer, and Shiri Lev-Ari. The role of social network structure in the emergence of linguistic structure. *Cognitive Science*, 44(8):e12876, 2020. doi: https://doi.org/10.1111/cogs.12876.

Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. In Francis R. Bach and David M. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 1530–1538. JMLR.org, 2015.

Matiss Rikters. Impact of corpora quality on neural machine translation. In Kadri Muischnek and Kaili Müürisep, editors, *Human Language Technologies - The Baltic Perspective - Proceedings of the Eighth International Conference Baltic HLT 2018, Tartu, Estonia, 27-29 September 2018*, volume 307 of *Frontiers in Artificial Intelligence and Applications*, pages 126–133. IOS Press, 2018. doi: 10.3233/978-1-61499-912-6-126.

Brian Roark, Richard Sproat, Cyril Allauzen, Michael Riley, Jeffrey Sorensen, and Terry Tai. The OpenGrm open-source finite-state grammar software libraries. In *Proceedings of the ACL 2012 System Demonstrations*, pages 61–66, Jeju Island, Korea, July 2012. Association for Computational Linguistics.

José Carlos Rosales Núñez, Djamé Seddah, and Guillaume Wisniewski. Comparison between NMT and PBSMT performance for translating noisy user-generated content. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, pages 2–14, Turku, Finland, 30 September – 2 October 2019. Linköping University Electronic Press.

José Carlos Rosales Núñez, Djamé Seddah, and Guillaume Wisniewski. Understanding the impact of UGC specificities on translation quality. In *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*, pages 189–198, Online, November 2021a. Association for Computational Linguistics. doi: 10.18653/v1/2021.wnut-1.22.

José Carlos Rosales Núñez, Guillaume Wisniewski, and Djamé Seddah. Noisy UGC translation at the character level: Revisiting open-vocabulary capabilities and robustness of char-based models. In *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*, pages 199–211, Online, November 2021b. Association for Computational Linguistics. doi: 10.18653/v1/2021.wnut-1.23.

Johann Roturier and Anthony Bensadoun. Evaluation of mt systems to translate user generated content. *Proceedings of Machine Translation Summit XIII, Xiamen, China*, pages 244–251, 2011.

Benoît Sagot and Pierre Boullier. SxPipe 2: architecture pour le traitement pré-syntaxique de corpus bruts. *Traitement Automatique des Langues*, 49(2):155–188, 2008.

Manuela Sanguinetti, Lauren Cassidy, Cristina Bosco, Özlem Çetinoglu, Alessandra Teresa Cignarella, Teresa Lynn, Ines Rehbein, Josef Ruppenhofer, Djamé Seddah, and Amir Zeldes. Treebanking user-generated content: a UD based overview of guidelines, corpora and unified recommendations. *CoRR*, abs/2011.02063, 2020.

Takayuki Sato, Jun Harashima, and Mamoru Komachi. Japanese-english machine translation of recipe texts. In Toshiaki Nakazawa, Hideya Mino, Chenchen Ding, Isao Goto, Graham Neubig, Sadao Kurohashi, Ir. Hammam Riza, and Pushpak Bhattacharyya, editors, *Proceedings of the 3rd Workshop on Asian Translation, WAT@COLING 2016, Osaka, Japan, December 2016*, pages 58–67. The COLING 2016 Organizing Committee, 2016.

Sergej Schmunk, Wolfram Höpken, Matthias Fuchs, and Maria Lexhagen. Sentiment analysis:

Extracting decision-relevant knowledge from ugc. In Zheng Xiang and Iis Tussyadiah, editors, *Information and Communication Technologies in Tourism 2014*, pages 253–265, Cham, 2013. Springer International Publishing. ISBN 978-3-319-03973-2.

Mike Schuster and Kaisuke Nakajima. Japanese and korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2012, Kyoto, Japan, March 25-30, 2012*, pages 5149–5152. IEEE, 2012. doi: 10.1109/ICASSP.2012.6289079.

Mike Schuster and Kuldip K. Paliwal. Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.*, 45(11):2673–2681, 1997. doi: 10.1109/78.650093.

Djamé Seddah, Benoît Sagot, Marie Candito, Virginie Mouilleron, and Vanessa Combet. The french social media bank: a treebank of noisy user generated content. In *COLING 2012, 24th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, 8-15 December 2012, Mumbai, India*, pages 2441–2458, 2012.

Rico Sennrich and Biao Zhang. Revisiting low-resource neural machine translation: A case study. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 211–221, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1021.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*, 2016.

Sofia Serrano and Noah A. Smith. Is attention interpretable? In Anna Korhonen, David R. Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2931–2951. Association for Computational Linguistics, 2019. doi: 10.18653/v1/p19-1282.

Hendra Setiawan, Matthias Sperber, Udhyakumar Nallasamy, and Matthias Paulik. Variational neural machine translation with normalizing flows. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the*

Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020, pages 7771–7777. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.694.

C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3): 379–423, 1948. doi: https://doi.org/10.1002/j.1538-7305.1948.tb01338.x.

Blaz Skrlj, Jan Kralj, Nada Lavrac, and Senja Pollak. Towards robust text classification with semantics-aware recurrent neural architecture. *Mach. Learn. Knowl. Extr.*, 1(2):575–589, 2019. doi: 10.3390/make1020034.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. MASS: masked sequence to sequence pre-training for language generation. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 5926–5936. PMLR, 2019.

Lucia Specia, Zhenhao Li, Juan Miguel Pino, Vishrav Chaudhary, Francisco Guzmán, Graham Neubig, Nadir Durrani, Yonatan Belinkov, Philipp Koehn, Hassan Sajjad, Paul Michel, and Xian Li. Findings of the WMT 2020 shared task on machine translation robustness. In Loïc Barrault, Ondrej Bojar, Fethi Bougares, Rajen Chatterjee, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Alexander Fraser, Yvette Graham, Paco Guzman, Barry Haddow, Matthias Huck, Antonio Jimeno-Yepes, Philipp Koehn, André Martins, Makoto Morishita, Christof Monz, Masaaki Nagata, Toshiaki Nakazawa, and Matteo Negri, editors, *Proceedings of the Fifth Conference on Machine Translation, WMT@EMNLP 2020, Online, November 19-20, 2020*, pages 76–91. Association for Computational Linguistics, 2020.

Matthias Sperber, Graham Neubig, Jan Niehues, and Alex Waibel. Neural lattice-to-sequence models for uncertain inputs. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 1380–1389, 2017.

Sara Stymne. Spell checking techniques for replacement of unknown words and data cleaning for haitian creole SMS translation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation, WMT@EMNLP 2011, Edinburgh, Scotland, UK, July 30-31, 2011*, pages 470–477, 2011.

Jinsong Su, Zhixing Tan, Deyi Xiong, Rongrong Ji, Xiaodong Shi, and Yang Liu. Lattice-based recurrent neural network encoders for neural machine translation. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pages 3302–3308, 2017.

Martin Sundermeyer, Tamer Alkhouli, Joern Wuebker, and Hermann Ney. Translation modeling with bidirectional recurrent neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 14–25, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1003.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112, 2014.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2818–2826. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.308.

E. G. Tabak and Cristina V. Turner. A family of nonparametric density estimation algorithms. *Communications on Pure and Applied Mathematics*, 66(2):145–164, 2013. doi: https://doi.org/10.1002/cpa.21423.

Esteban G. Tabak and Eric Vanden-Eijnden. Density estimation by dual ascent of the log-likelihood. *Communications in Mathematical Sciences*, 8(1):217 – 233, 2010. doi: cms/1266935020.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 1556–1566, 2015.

The New York Times. Facebook apologizes for vulgar translation of chinese leader's name, 2020. URL https://www.nytimes.com/2020/01/18/world/asia/facebook-xi-jinping.html.

Artem Timoshenko and John R. Hauser. Identifying customer needs from user-generated content. *Marketing Science*, 38(1):1–20, 2019. doi: 10.1287/mksc.2018.1123.

Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. Modeling coverage for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 76–85, Berlin, Germany, August 2016. Association for Computational Linguistics.

Raghavendra Udupa and Hemanta Kumar Maji. Computational complexity of statistical machine translation. In Diana McCarthy and Shuly Wintner, editors, *EACL 2006, 11st Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference, April 3-7, 2006, Trento, Italy*. The Association for Computer Linguistics, 2006.

Rianne van den Berg, Leonard Hasenclever, Jakub M. Tomczak, and Max Welling. Sylvester normalizing flows for variational inference. In Amir Globerson and Ricardo Silva, editors, *Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence, UAI 2018, Monterey, California, USA, August 6-10, 2018*, pages 393–402. AUAI Press, 2018.

Rob van der Goot and Gertjan van Noord. Modeling input uncertainty in neural network dependency parsing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 4984–4991, 2018.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc., 2017.

The Verge. The oxford dictionaries' word of the year is an emoji, 2015. URL https://www.theverge.com/2015/11/16/9746650/word-of-the-year-emoji-oed-dictionary.

Sandrine Wachs. Écriture numérique spontanée et variabilité : un écrit/oral à exploiter en Français Langue Étrangère (sensibiliser aux styles et à la prononciation). In Henry Tyne, Mireille Bilger, Paul Cappeau, and Emmanuelle Guerin, editors, *La variation en question(s) - Hommages à Françoise Gadet*, number 36 in Gramm-R, pages 237–254. Peter Lang, 2017.

Pidong Wang and Hwee Tou Ng. A beam-search decoder for normalization of social media text with application to machine translation. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, pages 471–481, 2013.

Daniel Watson, Nasser Zalmout, and Nizar Habash. Utilizing character and word embeddings for text normalization with sequence-to-sequence models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 837–843, 2018.

Andy Way. Machine translation: where are we at today? In Erik Angelone, Maureen Ehrensbergerdow, and Gary Massey, editors, *The Bloomsbury Companion to Language Industry Studies*. Bloomsbury Academic Publishing, 2019.

Warren Weaver. Translation. *Machine translation of languages: fourteen essays*, 38(1):15–23, 1949.

Zach Wood-Doughty, Nicholas Andrews, and Mark Dredze. Convolutions are all you need (for classifying character sequences). In Wei Xu, Alan Ritter, Tim Baldwin, and Afshin Rahimi, editors, *Proceedings of the 4th Workshop on Noisy User-generated Text, NUT@EMNLP 2018, Brussels, Belgium, November 1, 2018*, pages 208–213. Association for Computational Linguistics, 2018. doi: 10.18653/v1/w18-6127.

Felix Wu, Angela Fan, Alexei Baevski, Yann N. Dauphin, and Michael Auli. Pay less attention with lightweight and dynamic convolutions. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

Yingce Xia, Tianyu He, Xu Tan, Fei Tian, Di He, and Tao Qin. Tied transformers: Neural machine translation with shared encoder and decoder. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 5466–5473. AAAI Press, 2019. doi: 10.1609/aaai.v33i01.33015466.

Tim Z. Xiao, Aidan N. Gomez, and Yarin Gal. Wat zei je? detecting out-of-distribution translations with variational transformers. *CoRR*, abs/2006.08344, 2020.

Shuoheng Yang, Yuxin Wang, and Xiaowen Chu. A survey of deep learning techniques for neural machine translation. *CoRR*, abs/2002.07526, 2020.

Xiao Yang, Craig Macdonald, and Iadh Ounis. Using word embeddings in twitter election classification. *Inf. Retr. J.*, 21(2-3):183–207, 2018. doi: 10.1007/s10791-017-9319-5.

Yang Ye, Ming Zhou, and Chin-Yew Lin. Sentence level machine translation evaluation as a ranking. In Chris Callison-Burch, Philipp Koehn, Cameron S. Fordyce, and Christof Monz, editors, *Proceedings of the Second Workshop on Statistical Machine Translation, WMT@ACL 2007, Prague, Czech Republic, June 23, 2007*, pages 240–247. Association for Computational Linguistics, 2007.

Biao Zhang, Deyi Xiong, Jinsong Su, Hong Duan, and Min Zhang. Variational neural machine translation. In Jian Su, Xavier Carreras, and Kevin Duh, editors, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 521–530. The Association for Computational Linguistics, 2016. doi: 10.18653/v1/d16-1050.

Jingyi Zhang, Masao Utiyama, Eiichiro Sumita, Graham Neubig, and Satoshi Nakamura. Improving neural machine translation through phrase-based soft forced decoding. *Mach. Transl.*, 34(1):21–39, 2020. doi: 10.1007/s10590-020-09244-y.

Shengjia Zhao, Jiaming Song, and Stefano Ermon. Infovae: Balancing learning and inference in variational autoencoders. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019,*

The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019, pages 5885–5892. AAAI Press, 2019. doi: 10.1609/aaai.v33i01.33015885.

Hao Zheng, Zhanlei Yang, Wenju Liu, Jizhong Liang, and Yanpeng Li. Improving deep neural networks using softplus units. In *2015 International Joint Conference on Neural Networks (IJCNN)*, pages 1–4, 2015. doi: 10.1109/IJCNN.2015.7280459.