Understanding and Evaluating Unsupervised Cross-lingual Embeddings in the General and in the Clinical Domains

Félix Gaschi

▶ **To cite this version:**

Félix Gaschi. Understanding and Evaluating Unsupervised Cross-lingual Embeddings in the General and in the Clinical Domains. Computer Science [cs]. Université de Lorraine, 2023. English. NNT : 2023LORR0347 . tel-04584774

HAL Id: tel-04584774
https://hal.univ-lorraine.fr/tel-04584774

Submitted on 23 May 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSITÉ DE LORRAINE | BIBLIOTHÈQUES UNIVERSITAIRES

AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact bibliothèque : ddoc-theses-contact@univ-lorraine.fr
(Cette adresse ne permet pas de contacter les auteurs)

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4
Code de la Propriété Intellectuelle. articles L 335.2- L 335.10
http://www.cfcopies.com/V2/leg/leg_droi.php
http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm

UNIVERSITÉ DE LORRAINE

École doctorale IAEM Lorraine

Understanding and Evaluating Unsupervised Cross-lingual Embeddings in the General and in the Clinical Domains

THÈSE

présentée et soutenue publiquement le 14 Décembre 2023

pour l'obtention du

Doctorat de l'Université de Lorraine

(Informatique)

par

Félix Gaschi

Composition du jury

Président :	François Yvon, Professeur, Sorbonne Université, CNRS, ISIR
Rapporteurs :	François Yvon, Professeur, Sorbonne Université, CNRS, ISIR Anders Søgaard, Professeur, University of Copenhagen
Examinateurs :	En-Shiun Annie Lee, Professeur assistant, University of Toronto Asma Ben Abacha, Docteur, Microsoft
Invité :	François Plesse, Docteur, Posos
Encadrants :	Parisa Rastin, Docteur, Université de Lorraine Yannick Toussaint, Professeur, Université de Lorraine

Laboratoire Lorrain de Recherche en Informatique et ses Applications — UMR 7503
LORIA

Abstract

Labeled and unlabeled data are more often available in English than in other languages. In the clinical domain, non-English data can be even more scarce. Multilingual word representations have two properties that can help with this situation. The first one is multilingual alignment, where representations from different languages share the same latent space. More concretely, words that are translations of each other must have similar representations, which is useful for cross-lingual information retrieval. The second property is cross-lingual transfer learning: it allows a model to be trained on a supervised task in one language, and to provide good results for the same task in another language, without the need for any labeled data in that language.

This thesis addresses some gaps in the literature regarding the understanding of multilingual embeddings. It namely studies the link between multilingual alignment and cross-lingual transfer, showing that models like mBERT and XLM-R, that can perform cross-lingual transfer produce representations that have a stronger form of multilingual alignment than word embeddings that were explicitly trained for such alignment. It also finds a high correlation between cross-lingual transfer abilities and multilingual alignment suggesting that both multilingual properties are linked. This link allows to improve cross-lingual transfer for smaller models by simply improving alignment, which can allow them to match the performances of larger models but only for a low-level task like POS-tagging, due to the impact of fine-tuning itself on multilingual alignment.

While mainly focusing on the general domain, this thesis eventually evaluates cross-lingual transfer in the clinical domain. It shows that translation-based methods can achieve similar performance to cross-lingual transfer but require more care in their design. While they can take advantage of monolingual clinical language models, those do not guarantee better results than large general-purpose multilingual models, whether with cross-lingual transfer or translation.

Résumé

Un résumé en français de 10 pages est consultable en Annexe 5.

Les données, annotées ou non, sont plus souvent disponibles en anglais que dans d'autres langues. Dans le domaine clinique, les données non anglaises peuvent être encore plus rares. Les représentations, ou plongements lexicaux, multilingues peuvent avoir deux propriétés utiles dans cette situation. La première est l'alignement multilingue, où les représentations de différentes langues partagent le même espace latent. Plus concrètement, les mots qui sont la traduction l'un de l'autre doivent avoir des représentations similaires. La deuxième propriété est l'apprentissage par transfert cross-lingue : il permet à un modèle d'être entraîné sur une tâche supervisée dans une langue et de fournir de bons résultats pour la même tâche dans une autre langue, sans avoir besoin de données annotées dans cette langue.

Cette thèse aborde certaines lacunes de la littérature concernant la compréhension des représentations multilingues. Elle étudie notamment le lien entre l'alignement multilingue et le transfert cross-lingue, en montrant que les modèles, comme mBERT et XLM-R, qui peuvent effectuer ce transfert cross-lingue produisent des représentations qui ont une forme plus forte d'alignement multilingue que d'autres représentations qui ont été explicitement optimisées pour un tel alignement. Est également révélée la forte corrélation entre les capacités de transfert cross-lingue et l'alignement multilingue, ce qui suggère que ces deux propriétés sont liées. Ce lien permet d'améliorer le transfert cross-lingue pour les petits modèles en améliorant simplement l'alignement, ce qui peut leur permettre d'égaler les performances de grands modèles, mais seulement pour une tâche de bas niveau comme le POS-tagging, en raison de l'impact du fine-tuning lui-même sur l'alignement multilingue.

Tout en se concentrant principalement sur le domaine général, cette thèse évalue finalement le transfert multilingue dans le domaine clinique. Elle montre que les méthodes basées sur la traduction peuvent atteindre des performances similaires à celles du transfert multilingue, mais qu'elles nécessitent plus de soin dans leur conception. Et bien qu'elles puissent tirer parti de modèles linguistiques cliniques monolingues, ces modèles ne garantissent pas de meilleurs résultats que les larges modèles multilingues à usage général, que ce soit avec le transfert cross-lingue ou par traduction.

Table of Contents

Acronyms

BLI	Bilingual Lexicon Induction
CBoW	Continuous Bag-of-Words
CLG	Cross-lingual Generalization
CSLS	Cross-domain Similarity Local Scaling
CTL	Cross-lingual Transfer Learning
CWE	Contextualized Word Embedding
DSM	Distributional Semantics Model
LALM	Language-Agnostic Language Model
LM	Language Model
MA	Multilingual Alignment
MLM	Masked Language Modeling
mLM	Multilingual Language Model
mWE	Multilingual Word Embedding
NEL	Named Entity Linking
NER	Named Entity Recognition
NLI	Natural Language Inference
NSP	Next Sentence Prediction
OOV	Out-Of-Vocabulary
PCA	Principal Component Analysis
PEFT	Parameter-Efficient Fine-Tuning
POS-tagging	Part-of-Speech tagging
SWE	Static Word Embedding
TLM	Translation Language Modeling
WE	Word Embedding

Chapter 1

Introduction

This thesis aims to improve the understanding of multilingual embeddings motivated by applications in the clinical domain. It addresses some gaps in the literature on multilingual embeddings in general and then assesses whether multilingual embeddings learned in the general domain can be used in the clinical domain, namely on a Named Entity Recognition (NER) task.

One of the objectives of this thesis is to perform NER in the clinical domain. As shown in Figure 1.1, NER is the extraction of text spans representing different d types of entities. In the general domain, NER typically consists of extracting mentions of persons or geographical places. In the clinical domain, entities can be diseases, drugs, or therapeutic classes as in the figure.

Some language models can perform NER, but they need to be trained on already annotated examples first, before being able to perform the task on unseen examples. However, annotated data for specific tasks are scarce in languages other than English. Nevertheless, some multilingual language models have the ability to be trained on annotated examples from one language and perform well on examples from other languages. This ability is called Cross-lingual Transfer Learning (CTL). This thesis first focuses on studying CTL in the general domain, where multilingual data is mainly available, and it ultimately shows that general-domain multilingual models can be successfully applied in the clinical domain.

Another motivation for this thesis is to perform cross-lingual information retrieval. For example, once entities are extracted from sentences with NER, they can be normalized by finding the most similar standardized entities in a given lexicon. This process is called Named Entity Linking (NEL). In Figure 1.1, "Lyme borreliosis" is assigned to the Lyme disease code in a lexicon of diseases called CIM-10. Meanwhile, "antibacterials" is linked to the code corresponding to antibiotics in a different lexicon for therapeutic classes.

Similarly to NER, NEL can be performed in a cross-lingual setting. Cross-lingual NEL maps entities from different languages to the right term. Similar to any cross-lingual information retrieval task, cross-lingual NEL would require representations of texts that are shared across languages. In the case of a word embedding, which provides a different representation for each word, tokens

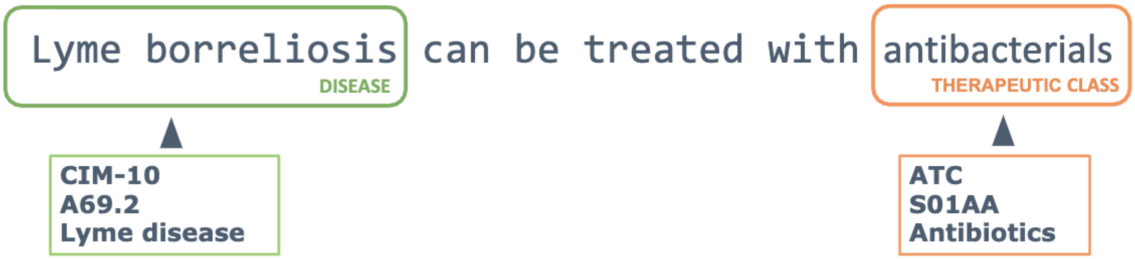


Figure 1.1 – Example annotated for clinical NER with extracted entities normalized with Named Entity Linking (NEL).

that are the translation of each other would typically need to have similar representations. This overlap of different languages in word representations is referred to as Multilingual Alignment (MA) throughout this thesis.

While the last chapter of this thesis contribution will focus on applications of multilingual embeddings to the clinical domain, the rest tries first to provide a better understanding of multilingual embeddings, and particularly of contextualized embeddings produced by models like mBERT (Devlin et al., 2019) and XLM-R (Conneau et al., 2020b). This thesis argues that a lot remains to be understood about those models, and brings some new elements of understanding about how they work.

The ability of mBERT and others to perform Cross-lingual Transfer Learning (CTL) is all the more surprising that they have not been explicitly trained for this property. They are able to perform a task in a given language without having seen any labeled data for that task and that language. The literature is rich in attempts to explain this phenomenon. A very recent review by (Philippy, Guo, and Haddadan, 2023) bears witness to it.

This thesis argues that trying to identify correctly what brings this ability to perform CTL is expensive. The ideal irrefutable experiment would be to find a property of the model on which a controlled experiment is performed: the model is pre-trained with and without this property and downstream evaluation verifies that the tested property is indeed what brings CTL. Pre-training a language model is expensive. There have been some attempts at smaller-scale experiments (Dufter and Schütze, 2020), but there is no guarantee that their conclusions generalize to a larger setting.

This thesis argues that we need first to gain a better understanding of the inner mechanics of existing models. It advocates for studying word representations produced by models like mBERT. This thesis believes that once a better understanding is gained about how those models work, maybe we will be able to study why they work. In that spirit, this thesis studies the link between Multilingual Alignment (MA) and Cross-lingual Transfer Learning (CTL). Some earlier generations of multilingual word embeddings, sometimes called static embeddings, are targeting multilingual alignment, while mBERT, XLM-R, and other Transformer-based models have demonstrated surprising CTL abilities, without explicitly aiming for multilingual alignment. Although the literature is not clear on the link between them, this thesis shows that models that can perform CTL usually have a good multilingual alignment of their inner representations. It also finds that this alignment is, in some respects, better than those of explicitly aligned multilingual embeddings. Furthermore, it finds a high correlation between multilingual alignment and CTL and further investigates the link between alignment and transfer, which differ depending on the downstream task involved.

Outline and contribution

The main contribution of this thesis is twofold: (1) providing a better understanding of multilingual word representations and (2) investigating how well cross-lingual transfer can perform outside of the general domain and more specifically in the clinical domain.

Chapter 2 explores different existing word-level language representations and how they can be extended to a multilingual setting, focusing on the distinction between multilingual alignment and cross-lingual transfer. Then, Chapter 3 focuses on the literature on static embeddings which usually aim for multilingual alignment while Chapter 4 focuses on larger contextualized language models that show surprising cross-lingual transfer learning abilities.

Chapter 5 gathers the main open questions raised in the previous chapters and provides a simple typology of multilingual representations according to the type of model used and the level of cross-lingual supervision, narrowing down the focus of the contribution.

The thesis first focuses on static embeddings. Chapter 6 shows that static embeddings can be trained in a language-agnostic manner like contextualized ones, but cannot provide a good multilingual alignment. It also shows that code-switching can have an impact on such language-agnostic training, but this impact is less noticeable on larger contextualized embeddings. Chapter 7 investigates the limitations of some unsupervised cross-lingual embeddings, showing that not all of them might be due to a lack of isometry.

Having shown the limitations of static embeddings, this thesis then provides a comparison of contextualized and static multilingual embeddings in terms of multilingual alignment (Chapter 8). It shows that contextualized embeddings, when evaluated with the right method contain representations that are better aligned than explicitly aligned static embeddings.

Chapter 9 then finds a high correlation between multilingual alignment and cross-lingual transfer, suggesting that improving alignment can actually improve cross-lingual transfer. However, it also shows that fine-tuning a multilingual model on a specific task in one language can surprisingly improve cross-lingual alignment, which gives some insight as to why improving alignment can, in some cases, fail to increase downstream cross-lingual performances.

While most contribution chapters do not focus on the clinical domain, Chapter 10 studies an application of cross-lingual transfer learning in the clinical domain, comparing it with methods based on translation. While this comparison has already been made in the general domain (Yarmohammadi et al., 2021), this last contribution chapter shows that the issue is different in a specific domain, as multilingual models are not domain-specific and some domain-specific monolingual modes can be leveraged for translation-based methods. Nevertheless, this thesis shows that cross-lingual transfer with general-domain multilingual models can provide better results than simply translating the training or test data and using a specialized model. Moreover, in a zero-shot cross-lingual setting, cross-lingual transfer has the advantage of working off the shelf.

Most of the work presented in this chapter is derived from the following works published during this thesis:

Gaschi, Félix, Parisa Rastin, and Yannick Toussaint (June 2021a). "Alignement non supervisé d'embeddings de mots dans le domaine biomédical." In: *CIFSD 2021 - Conférence Internationale Francophone sur la Science des Données*. Marseille/Virtuel, France. URL: https://hal.science/hal-03259987.

Gaschi, Félix, Parisa Rastin, and Yannick Toussaint (2021b). "Handling The Deviation From Isometry Between Domains And Languages In Word Embeddings: Applications To Biomedical Text Translation." In: *Neural Information Processing: 28th International Conference, ICONIP 2021, Sanur, Bali, Indonesia, December 8–12, 2021, Proceedings, Part II*. Sanur, Bali, Indonesia: Springer-Verlag, 216–227. ISBN: 978-3-030-92269-6. DOI: 10.1007/978-3-030-92270-2_19. URL: https://doi.org/10.1007/978-3-030-92270-2_19.

Gaschi, Félix, Alexandre Joutard, Parisa Rastin, and Yannick Toussaint (2022a). "Évaluation des propriétés multilingues d'un embedding contextualisé." In: *European Grid Conference*. URL: https://api.semanticscholar.org/CorpusID:247296214.

Gaschi, Félix, François Plesse, Parisa Rastin, and Yannick Toussaint (2022b). "Multilingual Transformer Encoders: a Word-Level Task-Agnostic Evaluation." In: *2022 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. URL: https://api.semanticscholar.org/CorpusID:250644290.

Gaschi, Felix, Patricio Cerda, Parisa Rastin, and Yannick Toussaint (July 2023a). "Exploring the Relationship between Alignment and Cross-lingual Transfer in Multilingual Transformers." In: *Findings of the Association for Computational Linguistics: ACL 2023*. Toronto, Canada: Association for Computational Linguistics, pp. 3020–3042. DOI: 10.18653/v1/2023.findings-acl.189. URL: https://aclanthology.org/2023.findings-acl.189.

Gaschi, Félix, Xavier Fontaine, Parisa Rastin, and Yannick Toussaint (July 2023b). "Multilingual Clinical NER: Translation or Cross-lingual Transfer?" In: *Proceedings of the 5th Clinical Natural Language Processing Workshop*. Toronto, Canada: Association for Computational Linguistics, pp. 289–311. DOI: 10.18653/v1/2023.clinicalnlp-1.34. URL: https://aclanthology.org/2023.clinicalnlp-1.34.

Chapter 2

Different Word Representations and their Extension to a Multilingual Setting

This thesis focuses on the generalization of word representation to a multilingual setting. More specifically it focuses on machine-learned language representations working at the word or sub-word level, and often called "word embeddings". It must therefore be noted that this thesis does not consequentially deal with sentence encoders in its core, although they might be evoked momentarily, and neither with generative language models, but rather with word embeddings, which are typically built for learning word representations.

Since some of those word embeddings are produced by language models, this chapter first provides a brief definition of language models and their link with word embeddings. It then describes existing word embedding approaches, emphasizing the distinction between static and contextualized embeddings. Finally, it makes the distinction between two potential abilities of multilingual versions of word embeddings: Cross-lingual Transfer Learning (CTL), which allows one to solve a task in one language by training in another, and Multilingual Alignment (MA), where different languages share the same latent representation.

2.1 Language models

A Language Model (LM) can be defined as a probability distribution over sequences of words. Rather than directly computing the probability of an entire sequence, a LM typically attempts to predict the probability of a word given its context. In the case where the context is all the preceding words, this conditional probability of a word w_i given all the previous ones $w_1...w_{i-1}$ allows to provide a probability over the entire sequence S of length L with the chain rule:

$$P(S) = P(w_L|w_1...w_{L-1})...P(w_i|w_1...w_{i-1})...P(w_2|w_1)P(w_1) \tag{2.1}$$

There are many ways to define the context and model the relationship between a word and its context. For example, some deep learning-based large language generative models are capable of generating human-like sensible answers to a large spectrum of possible queries. But simpler language models exist, like n-gram models, which compute a probability of a word only given the $n-1$ previous words, like the vanilla example shown in Figure 2.1. They are not powerful enough to generate long coherent texts like large generative models, but they can estimate an approximate likelihood of a given text to occur, and thus allow, for example, to improve the output of a translation model (Vogel, Ney, and Tillmann, 1996).

But this thesis is not necessarily interested in the ability of some language models to iteratively generate likely text, but rather in the knowledge of the language they can capture by being

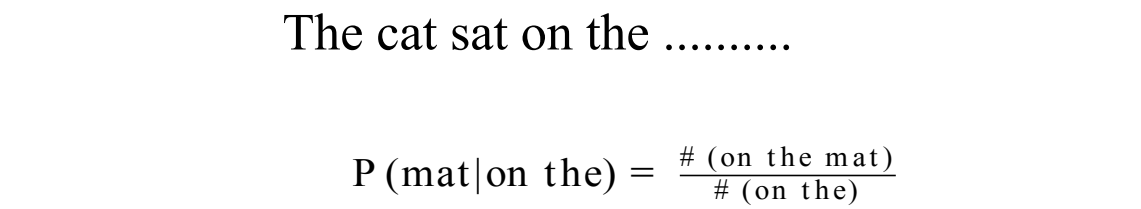


Figure 2.1 – **Vanilla trigram model**: the probability of the next word being "mat" is determined by the frequency counts of the trigram "on the mat" and the preceding bi-gram "on the". It must be noted that in this vanilla version, no smoothing is applied, which would cause an issue for new trigrams.

trained to learn the probability of a word given its context. Some of those language models build intermediary representations to capture the relationship between words. Some of those intermediary representations are the word embeddings in which this thesis takes interest.

Different LMs can have different contexts. Some models predict a word only from past tokens, which is required to generate new text. Others predict tokens also from future tokens. Because those latter models can "see in the future" they cannot be used for generating text. The utility of those models is not directly defined by their ability to predict a word given its context, but rather by the inner representations of language it has learned by doing so and the downstream tasks that those representations can solve.

2.2 Word embeddings

Language models that are trained to obtain those inner representations can be called "encoding language models" in opposition to generative ones. The language representations produced by such models are often called Word Embedding (WE). Depending on the nature of such a WE, the properties that are deemed interesting might vary.

Two main types of WE can be identified:

- Static Word Embedding (SWE) for which each word in a vocabulary is attributed a fixed embedding;
- Contextualized Word Embedding (CWE) where the embedding of a word varies with the context it lies in.

CWEs have the advantage of allowing different representations for different senses of a polysemous word. For example, the word "tie" can have a different embedding when found in the sentence "She is wearing a tie" or in the sentence "The match ended in a tie". However SWEs can also have their advantages. One of them is that once an embedding is trained, the representation of a given word can be obtained by a simple lookup table.

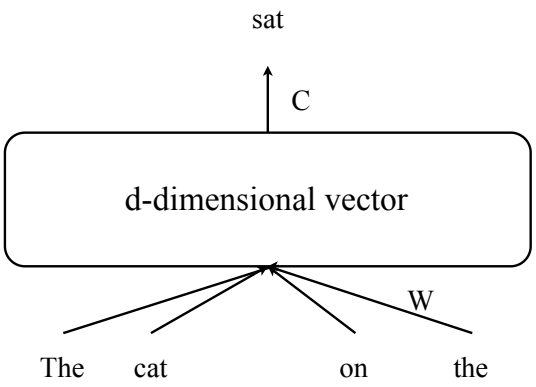
2.2.1 Static Word Embedding (SWE)

The notion of Word Embedding (WE) predates language models. WEs can be seen as a specific case of Distributional Semantics Model (DSM) also called Vector Space Model (VSM). A DSM aims at creating word representations from a corpus of text, relying on the observation that similar context implies similar meaning. In the words of the linguist John Rupert Firth:

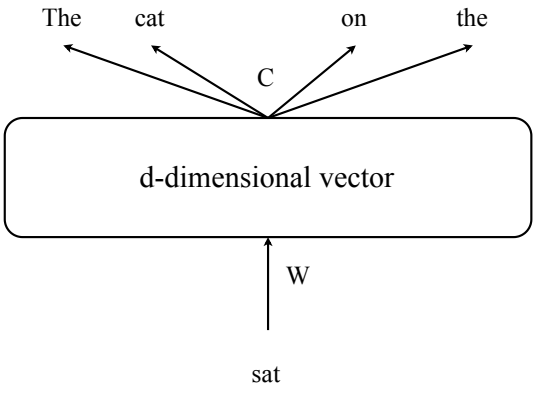
> You shall know a word by the company it keeps. Firth, 1959

Turney and Pantel (2010) give a comprehensive overview of DSMs based on counting-based methods. Observing words in a similar context allows one to build a co-occurrence matrix $M \in \mathbb{R}^{k \times n}$ with k the number of terms for which we build a representation and n the number of features or dimensions. These features can measure how likely a word is to appear in a given context (document, sentence, or sliding window of neighboring words). In this case, we are dealing with a term-context matrix. The features can also be co-occurrence frequencies, measuring how likely a word is to appear in the same context as another. In that case, we are talking about a term-term matrix. A term-context or a term-term matrix is sparse, discrete, and relatively large[1]. Thus existing DSMs often perform some dimensionality reduction technique on these term-term or term-context matrices.

With artificial neural networks, continuous word embeddings can be obtained in one step, without the need for dimensionality reduction. Continuous word embeddings are learned as latent representations of a shallow neural network trained to predict a word when being given one or several neighbors as input. Bengio, Ducharme, and Vincent (2000) introduced a continuous n-gram LM rather than a word embedding, which was shown to outperform existing discrete n-gram models in text generation tasks (Schwenk, 2007). Le, Allauzen, and Yvon (2012) showed that a similar model could beat discrete LMs on translation tasks. But these models were framed as "generative" n-gram LMs rather than as ways to obtain word embeddings such as a DSM. Mikolov et al. (2013) introduced word2vec, an algorithm that learns such continuous word embeddings with a shallow neural network with a lower computational cost than previous methods, and showed better results in a sentence completion task than a discrete n-gram model as well as a LSA[2]-based DSM. Word2vec actually proposes two methods: Continuous Bag-of-Words (CBoW) (Figure 2.2a) where a central word is predicted from its surrounding context and skip-gram (Figure 2.2b), where each context word is predicted from the central word.



(a) Continuous Bag-of-Words (CBoW): context words inside fixed windows (here of size 4), are each encoded by one-hot vectors of the size of the vocabulary (N). They are projected to a vector or size d (small w.r.t. N) with a matrix W of shape Nxd. Another matrix C projects back to a vector of size N. W and C are optimized such that "sat" has the highest coefficient in the output vector.



(b) Skip-gram: Similar to CBOW but predicts context words from the central one. For CBOW and Skip-gram, the lines of W give a vector representing each word in the vocabulary.

Figure 2.2 – Schema of how the two word2vec variants function.

One potential issue with WEs is the handling of Out-Of-Vocabulary (OOV) words. Indeed, words that were not seen during pre-training cannot be given a vector representation. To solve this issue, existing word embedding techniques have started to build representations of subwords. One early

[1]The initial word representations have as many dimensions as there are features.
[2]Latent Semantics Analysis (Deerwester et al., 1988)

example of that is FastText (Bojanowski et al., 2017) which builds upon skip-gram and CBoW by learning embeddings of character n-grams as well as whole words, thus OOV words can be represented by a linear combination of the representation of the character n-grams it contains.

SWEs are primarily built with the idea that words with similar meaning must have close representations. However, the distributional similarity that arises from sharing similar context can encapsulate a wide variety of relationships: synonymy and antonymy, but also values on a scale (like numbers or the name of months), hyponymy and hypernymy (like "horse" and "animal") (Turney and Pantel, 2010). There are also more complex relations that might be involved, like attributional similarity, where words for concepts that share salient features might have high distributional similarity, like "chair" and "horse", which both have four legs and a back.

Desirable properties of higher order can also be envisioned. Namely, the structure of embeddings can conform to some analogies. Mikolov, Yih, and Zweig (2013) show that the skip-gram model can learn a word embedding that roughly models analogies with simple linear algebra. For example, if we consider the addition and the subtraction between concepts to be modeled by respectively the addition and subtraction of their word embeddings then "King" - "Man" + "Woman" provides a vector that is close to the embedding of "Queen".

One widely-used evaluation method for SWEs is to measure the correlation between embedding similarity and human judgment. A data set for this evaluation consists of a set of word pairs for which human annotators gave a score indicating their semantic similarity. An SWE can then be evaluated by measuring the correlation between the human judgment and the similarity between the representations for all pairs of words. The similarity is usually expressed in terms of cosine similarity, which is written as:

$$\cos(u, v) = \frac{u^\top v}{||u||_2||v||_2} \tag{2.2}$$

For two vectors $u, v \in \mathbb{R}^d$ it is the scalar product between the normalized vectors $\frac{u}{||u||_2}$ and $\frac{v}{||v||_2}$.

However, SWEs have wider applications than reproducing human judgment for word similarity or inferring simple analogies. Namely, they can be used as a building block for machine learning models trained on a specific language processing task, which can be called **downstream task**. Before the advent of deep learning, DSMs could already be used as input features to traditional feature-based machine learning (Turian, Ratinov, and Bengio, 2010). It is not trivial to encode text in a deep learning model or even a simple linear model which both deal with dense representations. Word embedding allows the transformation of each word into a dense representation. Using a word embedding pre-trained on a large amount of unlabeled data as input to another model can help the model to perform well on words unseen during the training on a specific task (Collobert and Weston, 2008; Collobert et al., 2011).

Because SWEs can only capture the information about a single word, regardless of its context, it has some limitations. First of all, it cannot natively handle polysemy since the embedding is always the same, regardless of the context, for a polysemous word, which is a word that can bear a different meaning depending on the context. Moreover, because they encode only word-level information, SWEs are only a small building block of deep learning language processing models. Extensive labeled task-specific training data is thus necessary to learn adequate parameters of a deep learning model built on top of a SWE. In the following section, we describe Contextualized Word Embeddings (CWEs), which are obtained with more complex models and provide context-dependent word representations.

2.2.2 Contextualized Word Embedding (CWE)

While Contextualized Word Embeddings (CWEs) can be seen as improved Static Word Embeddings (SWEs) which can handle polysemy by taking the context into account, they are not exactly aiming at the exact same desirable properties. Static representations were explicitly expected to reflect some semantic similarity, and different SWEs are often compared using the correlation with the human judgment of some semantic similarity. Conversely, CWEs are more often compared on a benchmark of downstream tasks (Devlin et al., 2019). While SWEs can be used as the first layer of

a deep learning model, contextualized ones are themselves composed of several layers (including a learnable SWE as the first layer), and will compose the major part of the full model used for learning a specific downstream task. Devlin et al. (2019) typically add a single linear classification layer on top of the CWE when training it on a downstream task. This reduces the number of design choices needed to compare word embeddings on downstream tasks and might explain why CWEs are often directly compared on downstream tasks while static ones are evaluated on more intrinsic tasks like reproducing human judgment for word similarity or inferring simple analogies. Being a large part of the model, a CWE is usually not frozen when trained for a downstream task, which means that its weights are updated according to backpropagated gradients. The practice of such training on a downstream task is called fine-tuning. It must however be noted that the earliest attempts at CWEs like ELMo (Peters et al., 2018) were freezing the embedding for downstream tasks using them like traditional machine learning input features.

This pre-training-then-fine-tuning paradigm is called transfer learning. It gained popularity with models like BERT (Devlin et al., 2019) and its variants ALBERT (Lan et al., 2020), RoBERTa (Liu et al., 2019b), ELECTRA (Clark et al., 2020), SpanBERT (Joshi et al., 2020), DistilBERT (Sanh et al., 2019) or TinyBERT (Jiao et al., 2020). However, CWEs are still expected to have similar desirable properties as SWEs, as they can also be used for building representations of sentences or documents for information retrieval, which can then be performed by a simple nearest-neighbor search among the document representations (Reimers and Gurevych, 2019) or with more elaborate comparison between the query and the documents, like TSDAE (Wang, Reimers, and Gurevych, 2021) and ColBERT (Khattab and Zaharia, 2020) which are based on BERT.

There were some early contextualized word embeddings like ELMo (Peters et al., 2018) that were based on specific recurrent neural networks. However, most contextualized embedding, like BERT and its variants, rely on the Transformer architecture (Vaswani et al., 2017). It uses self-attention allowing it to train bigger models more easily thanks to a better parallelization than recurrent neural networks. This Transformer architecture was initially intended to perform machine translation and the first implemented Transformer was thus not explicitly a LM. But Transformer models have become so ubiquitous among LMs that they are sometimes almost considered to be synonymous with Language Models (LMs).

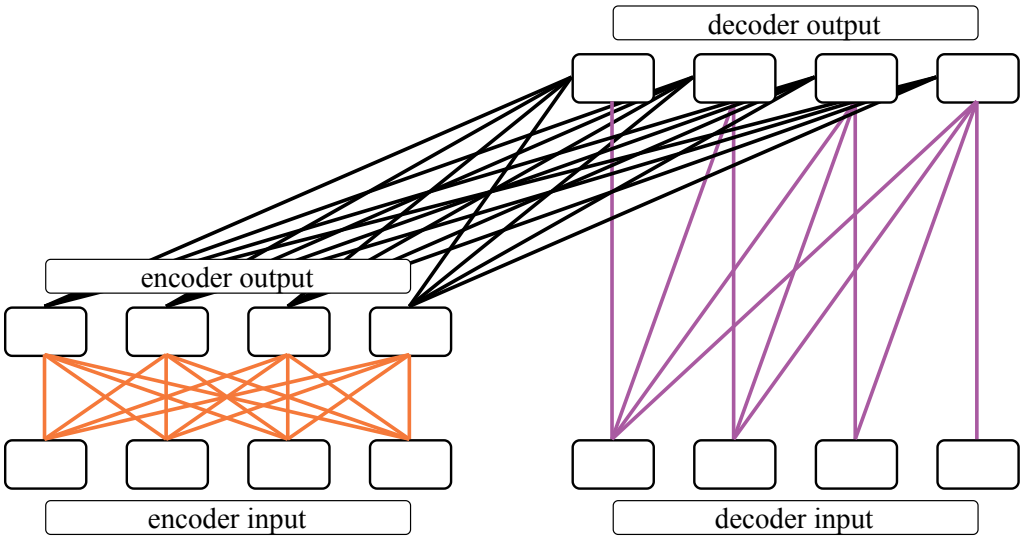
The original Transformer architecture is an encoder-decoder model. The encoder embeds an input sequence. The decoder produces the output sequence, where each new token can be predicted not only based on the encoder representations but also from the decoder representations of the previous tokens in the output, as shown in Figure 2.3a. Some LMs are keeping the whole architecture (encoder-decoder) but others use only the decoder or the encoder. They are called respectively decoder-only and encoder-only LMs. Prefix LMs (Figure 2.3d) also distinguish encoding and decoding, but both are handled in the same self-attention mechanism rather than by seperate components of the model (Raffel et al., 2020). The architecture is thus similar to a decoder-only or an encoder-only model and only the attention mask differs.

Contextualized embeddings are built with encoder-only LMs since they provide a bi-directional context. On the other hand, encoder-decoder and decoder-only models are rather used for building generative LMs.

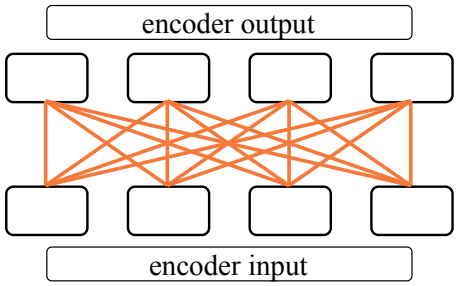
Transformer-based contextualized embeddings are composed of many learnable parameters, from hundreds of thousands to millions, and can be used as context for a given word all the other words from the text given as input. The input of those models can be relatively large, usually in the hundred tokens, despite a quadratic cost in compute with respect to the length.

The most widespread learning objective for contextualized LM is probably Masked Language Modeling (MLM) (Devlin et al., 2019). As shown in Figure 2.4, it is a very similar learning objective as CBoW, where the model is optimized to predict a word from its context. In practice, words are randomly selected in the pre-training data and then masked, and the model is trained to recover those words from the remaining others.

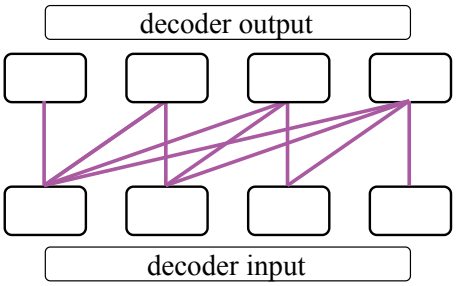
In BERT (Devlin et al., 2019), MLM is not only predicting masked-out words but also words that were randomly replaced by others. There also is an additional sentence-level objective called Next Sentence Prediction (NSP), where the model is trained on a binary classification task to determine whether the two input sentences are consecutive sentences or not related. The input



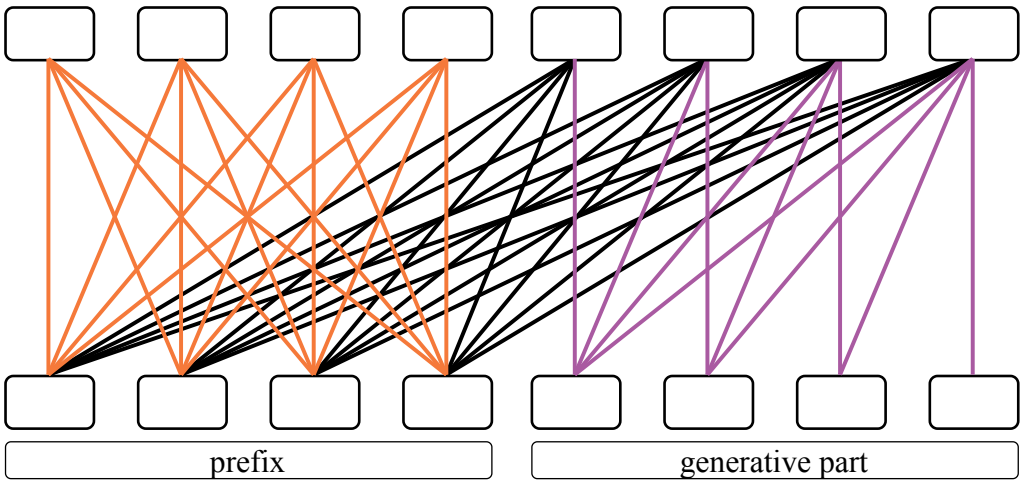
(a) Encoder-decoder



(b) Encoder-only



(c) Decoder-only



(d) Prefix LM

Figure 2.3 – Schema of the dependencies between the outputs and inputs in various architectures (encoder-decoder, encoder-only, decoder-only, and prefix LM). Each edge represents a dependency between an output and input unit, or between a decoder and an encoder output unit for the encoder-decoder architecture. **Orange** edges show encoder or encoder-like dependencies. **Violet** edges show decoder or decoder-like dependencies (causal). Black edges show connections between the encoder and decoder.

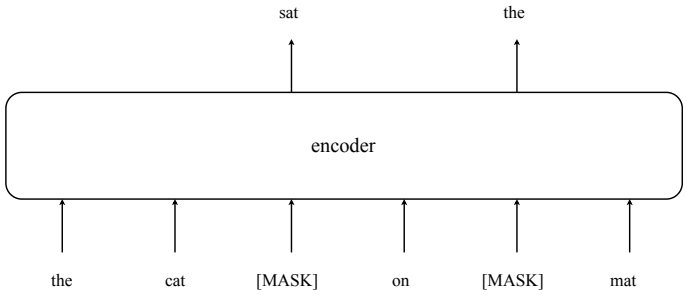


Figure 2.4 – Masked Language Modeling (MLM): A common learning objective for encoder-only Transformers. Random words are masked in the input and the model is trained to recover them.

for NSP is provided as "[CLS] first sentence [SEP] second sentence [SEP]" with special tokens "[CLS]" and "[SEP]". The prediction is made with a dedicated linear classifier applied to the output representation of the "[CLS]" token which can then be used to represent the whole sentence for downstream sentence-level tasks. However, Liu et al. (2019b) showed that the NSP objective can be detrimental and proposed RoBERTa which only relies on MLM.

MLM is sometimes framed as a denoising objective, where the model is trained to recover a corrupted input text. While generative models are often tasked with predicting the next word given the preceding ones, encoder LMs usually rely on such denoising objectives, which can take other forms than MLM with span recovering (Joshi et al., 2020) or even more complex objective using, for example, a smaller adversary LM which corrupts the input (Clark et al., 2020).

2.3 Cross-lingual abilities of language models

Having described the different sorts of word embeddings, how can they be extended to a multilingual setting?

A Multilingual Language Model (mLM) is a LM that can deal with multiple languages. In other words, it can compute the probability of a given sequence of words for different languages. But most multilingual models find their purpose by allowing some sort of shared knowledge between languages, otherwise a collection of monolingual LMs could be considered to be a mLM.

Two main types of cross-lingual abilities are extensively studied in the literature:

- Multilingual Alignment (MA)
- Cross-lingual Transfer Learning (CTL)

With Multilingual Alignment (MA), the aim is to build a Word Embedding (WE) which is consistent across languages. Therefore, if a monolingual embedding should represent words that have similar meaning with closer representations, then an aligned multilingual embedding should have this same requirement working across languages, which in practice would mean having each word to be close to its translation.

On the other hand, with Cross-lingual Transfer Learning (CTL), the aim is to build a mLM that, when trained on a downstream task in one language, can then be able to perform this task accurately in another language.

2.3.1 Multilingual Alignment (MA) for static embeddings

A word embedding has Multilingual Alignment (MA) when several languages share the same latent space. For SWE, it means that words that are the translation of each other must have close representations. For those static embeddings, MA can be evaluated with a task called Bilingual Lexicon Induction (BLI).

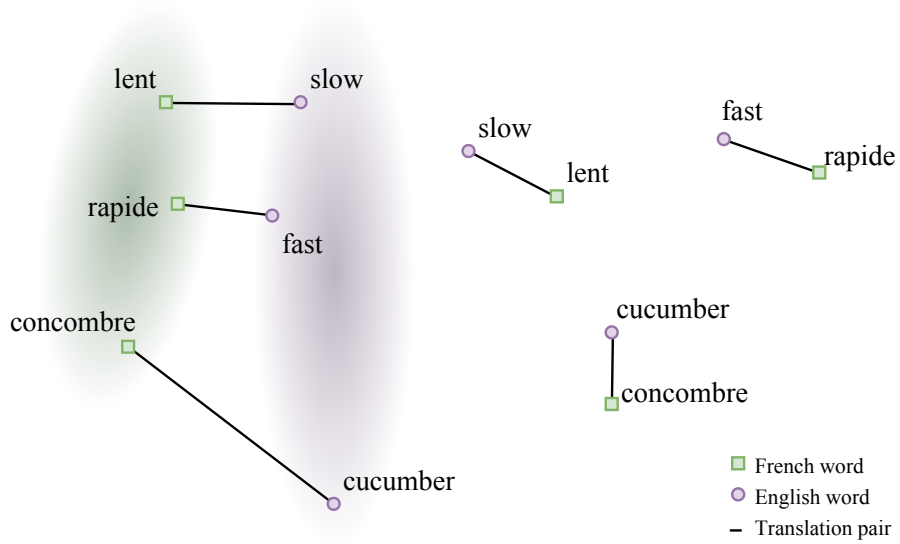
A bilingual dictionary defines pairs of words (w_1, w_2) between a source language and a target language, such that w_2 is a possible translation in the target language of the source word w_1. BLI is performed on a multilingual static embedding by inferring w_2 with the word in the target language vocabulary that has the closest embedding to w_1. The BLI accuracy is the fraction of source words in the bilingual dictionary for which the nearest neighbor is indeed the given translation.

To determine the closest neighbor, cosine similarity can be used, similar to monolingual embeddings when comparing with a human judgment of similarity (Equation 2.2). However, the high dimensionality of the representation causes some words to become hubs, i.e. words that are the nearest neighbor of many others. To alleviate this hubness problem (Dinu, Lazaridou, and Baroni, 2015), it was proposed to use the Cross-domain Similarity Local Scaling (CSLS) criterion (Lample et al., 2018), which is defined as follows:

$$\text{CSLS}(u, v) = 2\cos(u, v) - \frac{1}{K}\sum_{v' \in \mathcal{N}_V^K(u)} \cos(u, v') - \frac{1}{K}\sum_{u' \in \mathcal{N}_U^K(v)} \cos(u', v) \tag{2.3}$$

The cosine similarity cos between u and v is modified such that it removes the average similarity with the K nearest-neighbor across languages of u and v. More precisely, it takes the K nearest neighbors ($\mathcal{N}_V^K(u)$) in the target language of the source language embedding u and computes their average similarity with u. The same is done with v for nearest neighbors in the source language. Both averaged similarities are removed from the cosine similarity between u and v. Intuitively, this modified cosine similarity takes the density of representations around u and v into account.

2.3.2 Strong and weak multilingual alignments



(a) Weak alignment

(b) Strong alignment

Figure 2.5 – Toy example for weak and strong alignment. With weak alignment, representations for each language can be as far as possible from each other. The only requirement is that each word from language A is closer to its translation in language B than any word of language A. "lent" is closer to "slow" than "rapide" and "concombre". Strong alignment requires additional constraints. A word from language A must be closer to its translation than any other word from language A, like weak alignment, but also than any other word from language B. So "lent" is closer to "slow" than "rapide" and "concombre" but also than "fast" and "cucumber". In other words, weak alignment search for nearest neighbors in a bipartite graph, while strong alignment search for them in a fully connected graph.

A multilingual static embedding is deemed to be aligned if a word from language A is closer to its translation in language B than any other word in language A. However, certain downstream applications of multilingual embeddings might require a stronger definition of multilingual alignment. The difference between strong and weak alignment is motivated and explained in this section, and summarized in Figure 2.5.

The first interest in the literature for strong alignment comes from document retrieval (Roy et al., 2020). For multilingual document retrieval, it might be interesting to retrieve simultaneously documents in the same language as the query as well as documents from other languages. For example, a non-English native speaker who learned English as a second language might be interested in English documents as well as documents in their native language, to answer a query written in their native language.

Roy et al. (2020) were the first to propose a cross-lingual information retrieval dataset that targets such use cases. To the best of my knowledge, they are the first work to propose a distinction between strong and weak alignment.

Weak alignment is the usual way to perform and evaluate cross-lingual retrieval. Given cross-lingual representations in two languages A and B, for a query in language A, weak alignment looks for the answer only in language B. Multilingual static embeddings are always evaluated with weak alignment. In BLI, the query is a word in a given language and the search space is the vocabulary of another language. In information retrieval with datasets like XQuAD (Artetxe, Ruder, and Yogatama, 2020) or MLQA (Lewis et al., 2020), a query is a sentence in a given language while the pool of documents in which the answer must be retrieved is in another given language.

Strong alignment increases the search space and makes the task harder. The answer to a query in language A is searched among documents in language B, like weak alignment, but also in language A. This alignment measure is more strict and is relevant to some downstream tasks (Roy et al., 2020). However, this form of multilingual alignment was never used, to the best of my knowledge, to evaluate multilingual embeddings.

Open Question 2.1. *Are multilingual static embeddings competitive in terms of strong alignment?*

2.3.3 Cross-lingual Transfer Learning (CTL)

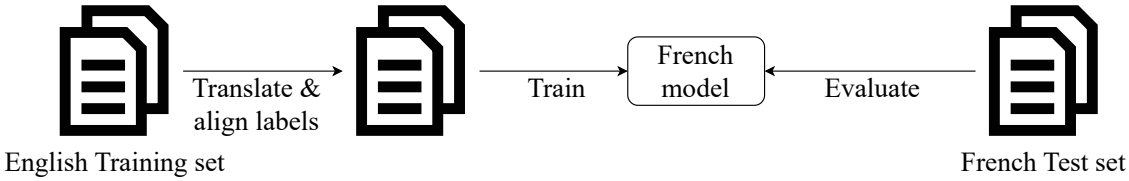
Contrary to static embeddings, monolingual contextualized embeddings are rather evaluated on downstream tasks than on intrinsic qualities of the produced representation. Similarly, multilingual contextualized embeddings are more often evaluated on cross-lingual downstream tasks than with some multilingual alignment metrics.

A typical setting of evaluation is Cross-lingual Transfer Learning (CTL). It is the same evaluation principle as fine-tuning on a downstream task, except that the performance is measured in another language than the training set. Benchmarks like XTREME (Hu et al., 2020b) have arisen to evaluate multilingual models on CTL, where models are often fine-tuned in English and evaluated in several other languages, for specific tasks like Natural Language Inference (NLI) (Conneau et al., 2018a), Named Entity Recognition (NER) or POS-tagging.

The motivation behind CTL is that fine-tuning data is often lacking in languages other than English. By training only in English, and evaluating in other languages, CTL alleviates the need for non-English training data. However, it could be argued that translating the English training dataset to the desired target language or translating the target data to English at inference could also solve the issue.

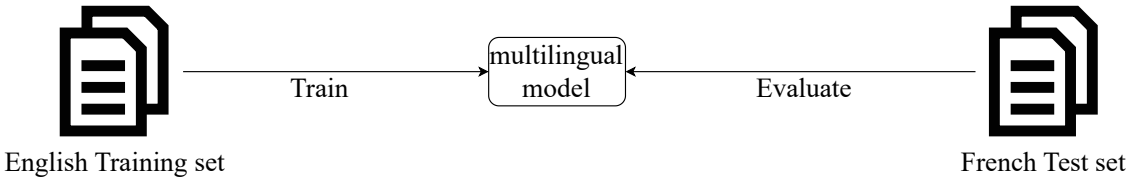
In this thesis, Cross-lingual Generalization (CLG) designates the fact of applying the knowledge obtained in one language to another, which can be obtained either with CTL or translation-based methods. The three CLG methods studied in this thesis are shown in Figure 2.6.

While well-designed translation-based methods might outperform existing mLM with CTL on sentence classification (Artetxe et al., 2023), the comparison is more complicated for word-level tasks such as Part-of-Speech tagging (POS-tagging) or NER. Yarmohammadi et al. (2021) show that neither translation-based methods nor CTL systematically outperforms the other after comparing them on several downstream tasks and languages.



(a) Translate-train

(b) Translate-test



(c) Cross-lingual Transfer Learning (CTL)

Figure 2.6 – Details of Cross-lingual Generalization (CLG) methods for a word-level task.

The additional difficulty with word-level tasks is that the labels applied to the original sentence must be placed on the right words in the translation. Additionally to a translation model for translating either the training or the test data, a word alignment model is also needed to align the labels between the original labeled sentences and their raw translations. A word alignment tool, like FastAlign (Dyer, Chahuneau, and Smith, 2013) or AWESOME-align (Dou and Neubig, 2021), produces a mapping of the words of one sentence with the words of its translation. FastAlign is a statistical tool, while AWESOME-align is based on the similarity of contextualized embeddings produced by mBERT, the multilingual version of BERT (Devlin et al., 2019).

To avoid any confusion, in this thesis, the expression "word alignment" will only be used for the task of mapping words between a sentence and its translation. On the other hand, any other use of "alignment" refers, except mentions of the contrary, to the Multilingual Alignment (MA) defined earlier.

2.3.4 Multilingual alignment for contextualized embeddings

Multilingual static embeddings are evaluated for Multilingual Alignment (MA), while contextualized embeddings are evaluated on their CTL abilities. One might then wonder whether static embeddings could be evaluated with CTL and contextualized ones with MA.

Similarly to the monolingual setting, evaluating static embeddings on downstream tasks is not straightforward, because word embeddings are only used as the first layer of more complex models. Hence there are many design choices to clarify before evaluating on downstream tasks, namely deciding what model to apply to the sequence of static embeddings. Nevertheless, Conneau et al. (2018a), who release XNLI, a dataset for evaluation CTL on Natural Language Inference (NLI), show that a baseline based on multilingual static embeddings does not give satisfactory results.

On the other hand, contextualized word embeddings could perfectly be evaluated for MA, and already have been. But despite the surprisingly efficient cross-lingual transfer of models like mBERT, there is no consensus on whether those multilingual models learn universal multilingual patterns, as a recent literature review states (Doddapaneni et al., 2021), as there are some additional challenges

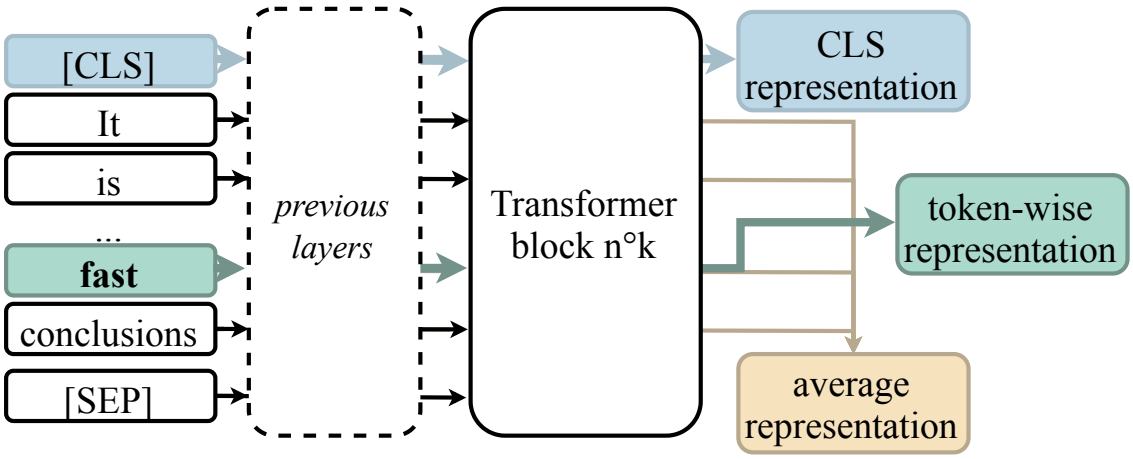


Figure 2.7 – Different sentence and token representations that can be extracted from contextualized embeddings.

to overcome when evaluating word-level alignment for contextualized embeddings. BLI is not directly usable on a contextualized embedding, because contextualized word embeddings need context to build a relevant representation of a word.

Evaluation of alignment of sentence representations is possible, and the XTREME benchmark for multilingual models contains two tasks for this, BUCC (Zweigenbaum, Sharoff, and Rapp, 2017, 2018) and Tatoeba (Artetxe and Schwenk, 2019), where the accuracy of nearest-neighbor search for sentence retrieval is measured. Usually, the representation of the first token is used to represent the sentence, especially for mBERT which uses this first token for the Next Sentence Prediction (NSP) training objective. It is the "CLS representation" in Figure 2.7. Singh et al. (2019) measure the similarity of these initial tokens for translated sentences and show that it decreases across layers, concluding that mBERT "is not an interlingua". However, this thesis argues that this is not conclusive because the similarity of translated pairs is not compared to those of random ones. The similarity between "[CLS]" tokens can do nothing but decrease across layers since the same token is compared. The first layer is a static embedding, leading to a similarity of 1. Pires, Schlinger, and Garrette (2019) perform a similar experiment but measure the accuracy of nearest-neighbor search similarly to Bilingual Lexicon Induction (BLI). The representation used is the average of all tokens in the sentence excluding special ones. It is the "average representation" in Figure 2.7. But the representations are also centered by language which removes a constant language-specific component. They find high retrieval accuracy for mBERT, but they do not report what the accuracy would be without language-wise centering. Conneau et al. (2020a) show that sentence representations obtained by pooling the representation of all tokens can be aligned to obtain high retrieval accuracy on cross-lingual sentence retrieval tasks, but then, they do not use the raw representations either. To the best of our knowledge, the Multilingual Alignment (MA) of sentence representations in contextualized embeddings is still to be evaluated, and compared for different pooling methods.

Open Question 2.2. *How good is the Multilingual Alignment (MA) of raw sentence representations obtained from contextualized word embeddings? What pooling method provides the best alignment?*

There is even less consensus about the alignment of word-level representations in the literature. One could imagine performing BLI with contextualized embeddings, but the issue is that those aptly named contextualized language models need context to build a representation of a word. Building the representation of singled-out words in a "[CLS] word [SEP]" template would seem artificial. A mLM might not have seen many - maybe any - singled-out words during its pre-training, and it

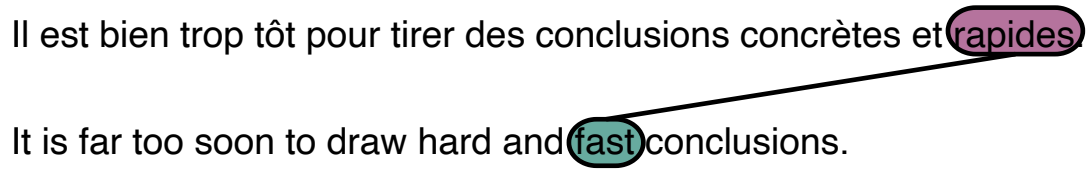


Figure 2.8 – Example of a pair of translated words extracted from a pair of translated sentences.

would not see any either during fine-tuning. Another possibility is to average several contextualized embeddings of the same word found in different sentences. But, across languages, one might not find equivalent contexts for words that are the translation of each other.

This leaves with one last possibility to build word-level representations of translated words with a contextualized embedding: applying word alignment on translated sentences. Given a translation dataset, a word alignment tool can be used to extract pairs of translated contextualized words as the one in Figure 2.8. Feeding each word of each pair with its context to a mLM allows one to extract a contextualized representation of it. It is the "token-wise representation" in Figure 2.7.

It must be noted that AWESOME-align (Dou and Neubig, 2021) relies on the similarity of those contextualized representations in mBERT to build a word alignment, and without any fine-tuning, it already provides better results than FastAlign. Thus, it can already be inferred that translated words extracted sentences have more similar representations than other pairs that can be drawn across those same sentences. For example, in Figure 2.8, the success of AWESOME-align shows that "fast" has a contextualized representation that is closer to "rapide" in the other sentence than any other word in that same French sentence.

However, a nearest-neighbor search in a whole sentence might be a lot easier than one in a typical BLI, since the search space is reduced to the words in the sentence, whereas the search space in BLI is the whole vocabulary of the target language.

Closer to something similar to BLI, several works have looked at the similarities of translated contextualized word pairs on a larger scale, extracting several of them in a whole translation dataset. Cao, Kitaev, and Klein (2020) compare the distribution of similarity of translated contextualized pairs with those of random pairs of words and show that they have a high overlap. But Efimov et al. (2023) show that this method can lead to incorrect conclusions. A high overlap in the distribution of similarities between related and random pairs means that sometimes random pairs can have higher similarities than related pairs. But since those pairs do not necessarily involve the same words, a high overlap does not mean that any word is closer to an unrelated one than to a related one. As an alternative, they propose to compare a related pair to its neighbors, which allows comparison of alignment before and after fine-tuning or re-alignment of representation but does not compare with BLI. This leads to the following question:

Open Question 2.3. *How do contextualized and static multilingual embeddings compare in terms of word-level multilingual alignment?*

Another issue with current attempts at studying multilingual alignment in contextualized embeddings is that they rely on FastAlign for extracting pairs of words. FastAlign can have an error rate that is quite important for some language pairs. This raises an additional question:

Open Question 2.4. *What is the impact of word alignment errors on the measure of multilingual alignment? How can we mitigate the impact of those errors?*

2.3.5 What about generative language models?

Static embeddings are often evaluated for multilingual alignment, and contextualized ones for CTL, but what about generative LMs?

Regarding cross-lingual transfer learning, generative LMs can be used to solve the same classification and sequence labeling tasks as encoding LMs by reformulating the tasks as text-to-text ones.

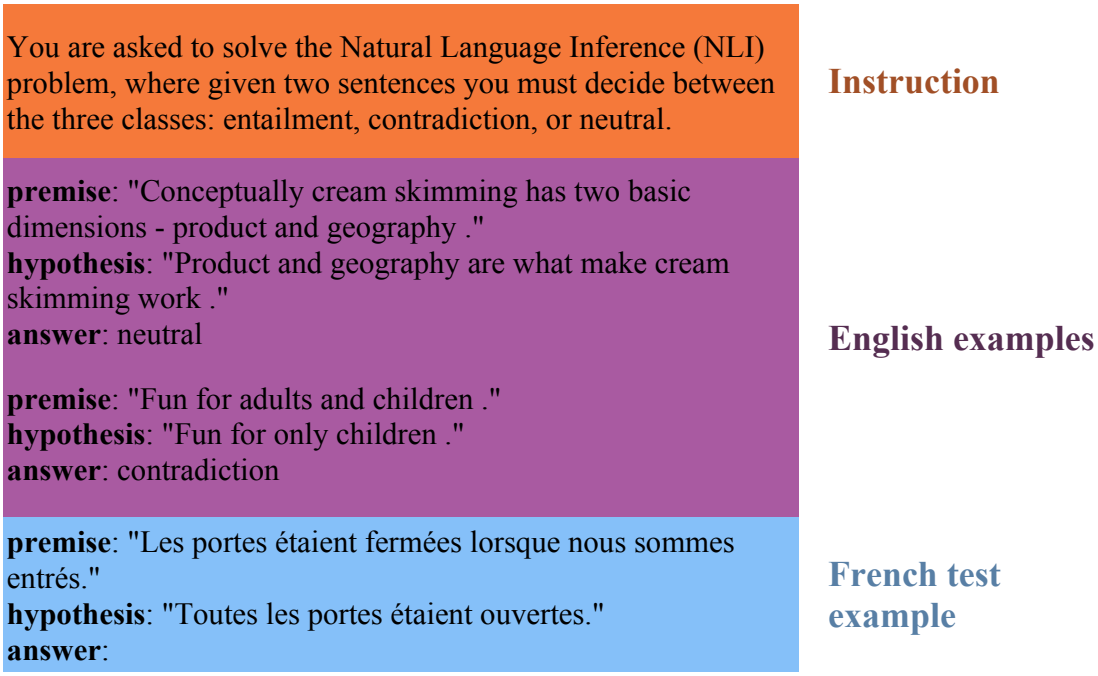


Figure 2.9 – Example of cross-lingual few-shot prompt for Natural Language Inference (NLI). In practice, the instruction can be more detailed and the examples can contain a verbalizer like "entailment, contradiction, or neutral?" just before the answer.

For example, the CoLA dataset (Warstadt, Singh, and Bowman, 2019) which classifies sentences according to their acceptability (i.e. whether they are making sense), can be transformed with the following instruction: "cola sentence: The course is jumping well" expecting the answer "not acceptable.". On those tasks, Ahuja et al. (2023) show that an encoder-decoder generative LM like mT5 (Xue et al., 2021), provides performances that are comparable with encoder-only LMs of comparable sizes: it provides better results than the smaller mBERT and worse ones than the larger XLM-R Large.

But larger decoder-only models of the GPT family (Brown et al., 2020; Radford and Narasimhan, 2018; Solaiman et al., 2019) and similar models are built for zero-shot or few-shot prompt learning. A typical cross-lingual few-shot prompt would then look like Figure 2.9. Ahuja et al. (2023) show that cross-lingual transfer with fine-tuning of contextualized embeddings largely outperforms any GPT-like model with zero-shot cross-lingual prompting.

However, contrary to encoding LMs, generative LMs can also be used for generation tasks, like summarization. In a cross-lingual setting, there are still a few issues with existing models. Models tend to produce incoherent texts (Rönnqvist et al., 2021), they suffer from some catastrophic forgetting (Vu et al., 2022), and more importantly, Xue et al. (2021) showed, when releasing mT5, that the model tends to generate text in the wrong language in a cross-lingual setting. Li and Murray (2023) investigate this issue, showing that fine-tuning a generative LM makes it more language-agnostic[3] explaining why generation might produce answers in the wrong language.

Regarding multilingual alignment, since generating texts requires some language-specific knowledge to avoid generating in the wrong language, it might be possible that representations built by the models are not that well aligned, at least in the decoder in an encoder-decoder architecture or the upper layers of a decoder-only models. That would explain why Li and Murray (2023) uses the encoder representations when measuring the cross-lingual alignment of sentence representations in mT5. But the question remains open for decoder-only models.

[3]Multilingual alignment of sentence representation in the encoder increases.

Open Question 2.5. *Are inner representations of multilingual generative LMs well aligned? What would be the difference between encoder-decoder and decoder-only architectures concerning multilingual alignment?*

Despite the latter open question, this thesis focuses on word embeddings and the distinction between static and contextualized ones. Eventually, it takes interest in applications for classification tasks in the clinical domain, rather than generation tasks. For all these reasons, only static and contextualized word embeddings are studied rather than generative LMs, which are left as a future avenue of research.

2.4 Recapitulation of definitions

We list here all the key concepts we defined above along with their definitions. This is intended as a small glossary to which the reader can go back if they feel unsure about the scope of some concept:

Language Model: an algorithm that defines a probability distribution over sequences of words and that gives an estimation of the likelihood of a unit of text given a specific context.

Word Embedding: Latent language representation that can be used as input to models for downstream tasks or as a representation of chunks of texts with nice properties, for example allowing retrieval of similar words by nearest-neighbor search.

Static Embedding: Word embedding for which the representation of a given word is not context-dependent.

Contextualized Embedding: Word embedding for which the representation of a given word varies with the context it lies in.

Multilingual Alignment: Property of a multilingual word embedding where a word and its translation must always have close representations, extending the desired properties of a monolingual embedding to a multilingual setting.

Word Alignment: mapping between the words of a sentence and those of its translation such that a word in one sentence is mapped to one of equivalent meaning in the translation.

Cross-lingual Generalization: The fact of using a training set in a given language to perform inference on test data for the same task in another language. This can be achieved either with translation or with cross-lingual transfer.

Cross-lingual Transfer: Fine-tuning a mLM on a source language training set and obtaining high accuracy in the target language for the same task.

Translate-train: Cross-lingual generalization methods where the training set is translated to a target language such that a model can be fine-tuned and evaluated on target language data. If the task is word-level, labels must also be projected with word alignment.

Translate-test: Cross-lingual generalization methods where a model is trained on the source language training set and inference is performed on target language by translating to source language first, then predicting the labels. Finally, if the task is word-level, the predicted labels must be projected onto initial target language examples with word alignment.

2.5 Conclusion

Language Models (LMs) can take many forms. This thesis differentiates generative LMs and encoding LMs. Generative LMs might be more powerful than encoding ones, as they can solve generative tasks like summarization on top of classification tasks that both encoding and generative models can solve.

However, the application of those generative LMs to a cross-lingual setting seems for now relatively anecdotical, while encoding LMs and the word embeddings already provide desirable multilingual properties that are truly cross-lingual like Multilingual Alignment (MA) and Cross-lingual Transfer Learning (CTL).

The following two chapters first provide more detailed background on how previous works proposed multilingual shallow and static embeddings with MA in mind, and then describe how contextualized and deep mLMs show strong CTL abilities, often without any explicit cross-lingual training signal.

Chapter 3

Unsupervised Multilingual Static Embeddings

A Word Embedding (WE) represents words with points in a metric space such that words with similar meanings will have close representations. A Multilingual Word Embedding (mWE) generalizes this requirement across languages. Words from different languages should have close representations if they are translations of each other. This leads to a whole line of works where mWEs are built using monolingual static embeddings and aligning them together, such that pairs of translated words coincide. This chapter describes such static mWEs with a focus on those built in an unsupervised manner and their limitations. For a more comprehensive overview of those methods as well as supervised ones, the reader might refer to the book from Sogaard et al. (2019) on which this chapter relies.

3.1 General principle

Most works on multilingual static word embeddings rely on word2vec or a variant of it, namely FastText (Bojanowski et al., 2017), which is the skip-gram model augmented with representations of character n-grams allowing us to build representations for words that were originally out of vocabulary.

As for DSMs built with non-neural approaches, sometimes referred to as "context-counting" methods, it must be noted that multilingual versions of them were also imagined. The features used to build a term-document or term-term matrix can be language-independent, using a bilingual lexicon, or parallel sentences. Sogaard et al. (2019) give an account of these "pre-embedding" methods[4].

3.1.1 From monolingual to multilingual embeddings

In a multilingual word embedding, not only do we need words from the same language to be close to each other if they have similar meanings but we also need words from different languages to go by the same rule, namely to be the closest possible to each other if they are a translation pair. An ideal multilingual word embedding should look like the toy example in Figure 3.1.

A multilingual word embedding must meet several criteria. First, it must be consistent for each language it is representing. This means that if we restrict the vocabulary of a multilingual embedding to a single language, the resulting embedding must be a good monolingual embedding. Second, the multilingual word embedding must provide a good Multilingual Alignment (MA). To put it plainly, words that are translations of each other must be close to each other.

[4]Chapter 4 of Sogaard et al. (2019)

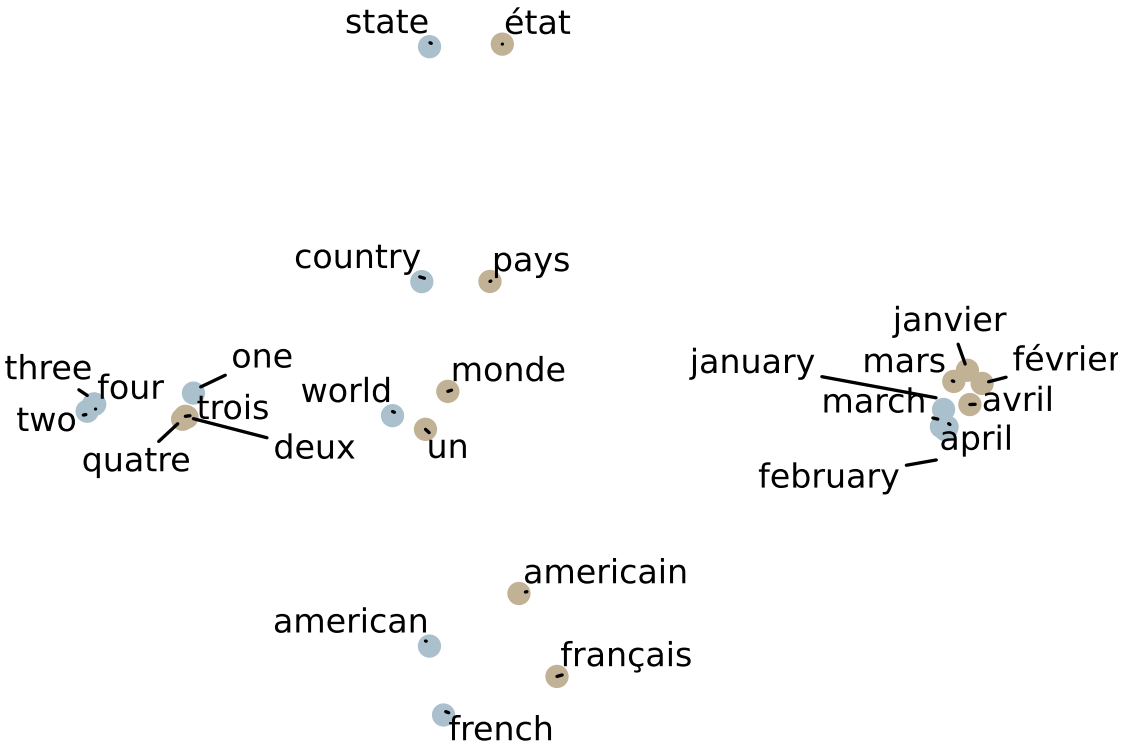


Figure 3.1 – Toy example of a multilingual word embedding. Obtained with the embeddings of a few pairs of translated words from FastText (Bojanowski et al., 2017) aligned with the RCSLS method from (Joulin et al., 2018). The 2D projection is given by the second and third components of the Principal Component Analysis (PCA).

These criteria can be formulated as the different terms of a loss function that a given multilingual embedding must optimize. Following the formalism of Sogaard et al., 2019, the loss can be written as :

$$J = \sum_{i=1}^{n} \mathcal{L}_i + \sum_{i=1}^{n} \sum_{j=1 \neq i}^{n} \Omega_{ij} \tag{3.1}$$

There are n different languages, $\mathcal{L}_i$ is the loss function for the monolingual consistency of the embedding for the i-th language, and Ω_{ij} is the loss function which targets MA between the i-th and j-th languages.

For a bilingual word embedding, which is the most studied case (Sogaard et al., 2019), the equation boils down to:

$$J = \mathcal{L}_1 + \mathcal{L}_2 + \Omega_{1,2} \tag{3.2}$$

Solving the bilingual case allows one to find an alignment for $n > 2$ languages with several strategies. The most straightforward one is to choose a pivot language and align each of the other $n - 1$ languages to it by solving each corresponding bilingual case. However, using a pivot language can degrade the quality of the indirect alignment between non-pivot languages, and finding an optimal multilingual alignment for more than two languages is not trivial given only a bilingual alignment method (Alaux et al., 2018). This is another instance where static embeddings fall short of contextualized ones which can be trained on hundreds of languages.

3.1.2 Existing methods and their similarities

Monolingual ($\mathcal{L}$) and multilingual (Ω) loss terms can be optimized jointly or sequentially. Existing methods can be distributed in three categories according to how the cross-lingual signal is injected:

- **Explicitly joint approaches**: the monolingual losses $\mathcal{L}_i$ and the explicit cross-lingual loss Ω are jointly minimized.
- **Pseudo-mixing approaches**: the learning is also joint, but instead of explicitly defining a cross-lingual loss, cross-lingual information is injected in a monolingual training by replacing words with their translation in the training data.
- **Mapping-based approaches**: the optimization is done sequentially. First monolingual embeddings are learned by optimizing $\mathcal{L}_i$ separately for each language. Then, embeddings are mapped together with a cross-lingual objective.

The three categories of approaches were shown to be roughly equivalent. More precisely, a specific pseudo-mixing approach, as well as a mapping-based method, were shown to be equivalent, in the limit, to a joint approach (Sogaard et al., 2019).

3.1.3 From supervised to unsupervised methods

While the three approaches are usable with cross-lingual supervision, relying on translated sentences or bilingual dictionaries, mapping-based approaches are the only ones, to the best of my knowledge, that were applied in an unsupervised cross-lingual setting.

Fully unsupervised methods might have been made easier by the smaller search space that goes with mapping-based methods. Indeed, those methods need to only learn a mapping between existing monolingual embeddings, while joint and pseudo-mixing approaches need to learn the whole embedding for each language. Nevertheless, this raises the following question:

Open Question 3.1. *While there are unsupervised mapping-based approaches, Is it possible to design pseudo-mixing or explicitly joint approaches in an unsupervised setting?*

It must be noted that Marchisio et al. (2022) propose IsoVec, which might be the only unsupervised method that is not exactly a mapping-based approach. IsoVec does not fit the categorization of the previous section: being given a single monolingual embedding in a source language, it learns one in a target language from scratch by jointly optimizing for a monolingual loss for the target language and a cross-lingual loss. It is not an explicitly joint approach, because the monolingual embedding in the source language is learned beforehand and not jointly. It is not a mapping-based approach either, because it optimizes jointly for cross-lingual alignment and monolingual consistency in the target language.

IsoVec is akin to an explicitly joint approach and demonstrates that unsupervised methods must not necessarily be mapping-based ones. However, IsoVec being the only exception to date, the rest of this thesis focuses on unsupervised mapping-based approaches when it comes to static embeddings.

3.2 Unsupervised mapping-based methods and isometry

Mapping-based approaches often rely on the assumption that monolingual embeddings from different languages have a similar structure: this is called the "isometry assumption".

This section shows that this assumption is supported by some observations and that there is some intuition about why it can hold. It also enumerates the practical reasons that require the isometric assumption. Finally, it deals with the limitations of such an assumption, as there are many cases in which this hypothesis does not seem to hold.

3.2.1 The isometry assumption

Mapping-based approaches gained traction after word2vec was introduced (Mikolov et al., 2013). It was noticed that its embedding of similar words in different languages could be aligned by a linear transformation (Mikolov, Le, and Sutskever, 2013), suggesting that a linear mapping can be learned between monolingual embeddings without having to learn it jointly with the monolingual objective.

To preserve the shape of monolingual embeddings and avoid losing what was learned during the monolingual training, the mapping applied to an embedding must be distance-preserving. Such a transformation is called isometry.

Definition 3.1. *Let (X, d_X) and (Y, d_Y) be two metric spaces. A map $f : X \mapsto Y$ is an isometry if for any $(x_i, x_j) \in X^2$, we have:*

$$d_X(x_i, x_j) = d_Y\left(f(x_i), f(x_j)\right) \tag{3.3}$$

The existence of such an isometry between two given metric spaces is not guaranteed. Therefore, mapping-based methods that learn an isometric mapping are based on the assumption that the monolingual embeddings are isometric.

Definition 3.2. *Two metric spaces (X, d_X) and (Y, d_Y) are isometric if and only if there exists an isometry f that maps each element of X to an element of Y.*

The success of those isometric mapping-based approaches rests on this "isometric assumption" that monolingual word embeddings learned for different languages are, at least approximately, isometric. The fact that the "isometric assumption" holds can seem surprising but Miceli Barone (2016) hypothesize that if the monolingual corpora, on which each monolingual embedding is learned, convey similar information in similar contexts, covering the same topics, then there could exist an approximate isometry between the random processes that allow to generate a word from its context for each language. By extension, this approximate isometry might exist between word embeddings trained to predict a word from its surroundings.

Fortunately, the success of such mapping-based approaches, supervised and unsupervised, seems to suggest that the assumption holds, at least in some settings. The next section shows that this assumption is helpful in several ways for mapping-based methods, particularly for unsupervised ones.

3.2.2 How isometry enables mapping-based approaches

An isometric mapping preserves distances Mapping-based approaches map together embeddings that are already trained for monolingual objectives. Therefore, the learned mapping must not degrade the monolingual consistency of the initial embeddings. By preserving distances between embeddings of a same language, an isometric mapping allows to retain the monolingual consistency.

Enforcing isometry simplifies the learning of the mapping In mapping-based methods, the mapping is often linear. And a linear isometry between two Euclidean spaces of the same dimension produces an orthogonal matrix, a square matrix whose transpose is its own inverse. The orthogonal mapping with a bilingual dictionary of N pairs. This dictionary can be provided in a supervised setting. In the unsupervised setting, this dictionary must also be learned, as will be explained later. In either cases, learning a mapping from a given dictionary can be formalized, following the notations of Lample et al., 2018, as:

$$W^* = \arg\min_{W \in \mathcal{O}_d} ||AW - B||_F^2 \tag{3.4}$$

$A \in \mathbb{R}^{N \times d}$ and $B \in \mathbb{R}^{N \times d}$ are the embeddings in $\mathbb{R}^d$ of the N entries in the bilingual dictionaries for the respective languages. Procrustes analysis (Schönemann, 1966) gives an exact solution to this problem: given the Singular Value Decomposition (SVD) of $A^\top B = USV^\top$, the solution is $W^* = UV^\top$. The isometric constraint thus provides a nice analytical solution to learn the mapping. Moreover, it can act as a regularization method as some methods reported better and more stable results when enforcing isometry (Lample et al., 2018).

3.2.3 Isometry allows for fully unsupervised methods

In a fully unsupervised setting, there is no training bilingual dictionary available. The parameters to learn are not only a mapping between entries of a bilingual dictionary but also the bilingual dictionary itself. The problem in Equation 3.4 becomes the following:

$$W^*, P^* = \arg\min_{W \in \mathcal{O}_d, P \in \mathcal{P}_N} ||XW - PY||_F^2 \tag{3.5}$$

In the supervised setting, W was mapping together conveniently ordered embeddings like A and B in Equation 3.4, whereas, here, W maps the whole monolingual embeddings X and Y in $\mathbb{R}^{N \times d}$. And since monolingual embeddings are provided in no particular order, a permutation matrix P must be learned to rearrange the lines of the matrix Y such that each word is aligned with its translation.

To tackle this double optimization problem, existing approaches have used iterative self-learning. It is a framework that was initially proposed to reduce the size of the training bilingual dictionary needed (Artetxe, Labaka, and Agirre, 2017). The idea is that, given a small initial dictionary, a first mapping could be learned with Procrustes like in a supervised method, but this mapping can then be refined by inferring a new larger bilingual dictionary from it, and then iteratively learning a new mapping and a new dictionary, until some convergence criterion is met. At each iteration, a mapping is learned with Procrustes from the previously learned bilingual dictionary. On the other hand dictionary inference can take many forms. Some methods use optimal transport like in the Wasserstein-Procrustes method (Grave, Joulin, and Berthet, 2018). But simpler heuristics like a nearest-neighbor search can also be used (Artetxe, Labaka, and Agirre, 2018a).

While iterative self-learning has allowed for a reduction in the size of the seed dictionary in a supervised setting (Artetxe, Labaka, and Agirre, 2017), it alone does not allow to learn a cross-lingual mapping in a fully unsupervised manner. Indeed by learning each new dictionary from an existing mapping and each new mapping from an existing dictionary, it means that a seed dictionary or an initial mapping must be learned in another way first.

To get rid of the need for any training seed dictionary, fully unsupervised mapping-based approaches have relied on various initialization approaches, usually relying on the isometry assumption.

The initialization approach is sometimes called seed dictionary induction (Sogaard et al., 2019) since the aim is to produce an initial seed dictionary, or an initial mapping, which ultimately allows one to learn a full dictionary.

Existing fully unsupervised mapping-based methods rely on either of the three approaches for seed dictionary induction:

- **Approximate alignment with PCA**: where PCA is performed as a dimensionality reduction technique and two languages are expected to share similar principal components;
- **Adversarial learning**: were a mapping is trained to fool a discriminator, itself trained to predict the language of an embedding from its (transformed) coordinates.
- **Graph-matching**: where the initialization is seen as a graph-matching problem between monolingual neighborhood graphs.

Approximate alignment with PCA. If two embeddings are linearly isometric and if they are anisotropic, applying PCA should align them. Indeed, if two embeddings are linearly isometric then they should only differ by a rotation. Projecting them to their principle axis of variation, as the PCA does, should align them. This technique is often used for cloud point matching (Daras, Axenopoulos, and Litos, 2012). Hoshen and Wolf (2018) used it as a seed induction method.

Adversarial learning. Generative Adversarial Networks (GAN) (Goodfellow et al., 2014) can be used to learn an initial mapping. In a vanilla GAN, two models are optimized adversarially: a generator g and a discriminator D. g allows to generate embeddings while D is trained to tell generated embeddings apart from given non-generated ones. Simultaneously, g is trained to fool D. In the case of seed dictionary induction (or rather seed mapping induction), the generator learns a mapping W to align the embedding X with another one Y. D is a feed-forward neural network trained to tell XW and Y apart. Formally, by denoting w as the learnable weights of D, the GAN optimizes the following loss:

$$\min_{W} \max_{w} E\left[\log D_w(Y) + \log\left(1 - D_w(XW)\right)\right] \tag{3.6}$$

Lample et al. (2018) use this method to learn an isometric mapping W that can then be refined through self-learning. Zhang et al. (2017b) uses a different version called Wassertein-GAN where the discriminator attempts to estimate the Wasserstein distance between Y and XW.

Graph-matching. The exact graph-matching problem can be defined as follows:

Definition 3.3. *Given two graphs $G_i = (V_i, E_i)$ and $G_j = (V_j, E_j)$ with set of vertices V_i and V_j such that $|V_i| = |V_j|$, the exact graph matching, if it exists, is the bijective function $f : V_j \to V_i$ such that $(u, v) \in E_j$ if and only if $(f(u), f(v)) \in E_i$.*

In other words, an exact graph matching is an isomorphism between two graphs. If we define the distance between vertices as the shortest path between them, the exact graph matching becomes an isometric function between the sets of vertices for this distance.

For two given graphs, an exact isomorphism rarely exists. Moreover, finding the exact graph-matching is an NP-complete problem. In practice, the seed dictionary induction methods are usually a convex relaxation of the graph-matching problem (Grave, Joulin, and Berthet, 2018) or a graph-matching-inspired heuristics where each word is represented by the sorted distance to all the other words. A nearest-neighbor search among those representations can then provide the seed dictionary. Such graph-matching-inspired heuristic is proposed by the VecMap method (Artetxe, Labaka, and Agirre, 2018a). It follows the following steps, given two embeddings X and Y:

- Using the Singular Value Decomposition $X = USV^\top$, the square root of the matrix of pairwise similarities $M_X = XX^\top$ is computed as $\sqrt{M_X} = USU^\top$. According to VecMap authors, it should thus be "closer in nature" to the original embedding than M_X. $\sqrt{M_Y}$ is computed in a similar manner.

- Sort the values in each rows of $\sqrt{M_X}$ and $\sqrt{M_Y}$, giving $X' = \text{sorted}(\sqrt{M_X})$ and $Y' = \text{sorted}(\sqrt{M_Y})$
- A seed dictionary can be obtained by a nearest-neighbor search between X' and Y'.

At first sight, this might seem unrelated to graph-matching, but the reader must note that M_X and M_Y are pairwise similarity matrices and can be seen as adjacency matrices of two weighted undirected graphs. The exact graph matching problem is equivalent to finding the optimal permutation of row and column indices for matching M_X and M_Y. Because this approach is intractable in practice, VecMap resorts to sorting each the values in each row of the two matrices (or rather their square roots). This boils down to representing each word by the sorted distances to all other words. A nearest-neighbor search among those representations thus matches vertices involved in similar edges.

Graph-matching can also be linked to optimal transport. For example, Alvarez-Melis and Jaakkola (2018) formulate the problem as finding a matching between two probability distributions rather than between two graphs.

Towards robust seed dictionary induction. methods based on PCA or Adversarial learning suffer from instability issues (Sogaard et al., 2019). However, Lample et al. (2018) observed that enforcing the approximate isometry of the mapping for their adversarial method empirically makes the training more stable, but still obtained a BLI accuracy that largely varies with different initializations.

On the other hand, graph-matching heuristics have been used successfully as a more stable seed dictionary induction technique with VecMap (Artetxe, Labaka, and Agirre, 2018a).

While they do not require an exact isomorphism between the underlying graphs, graph-matching-inspired methods still need the initial monolingual embeddings to be fairly isometric. Indeed they are trying to create a seed bilingual dictionary by matching together embeddings that have the most similar neighborhood, or the most similar set of distances with the other embeddings.

The isometry assumption does not only provide a nice closed-form solution for supervised alignment or in iterative self-learning, but it is instrumental to the success of existing seed dictionary induction techniques. In practice, most monolingual embeddings are only loosely isometric, and the seed dictionary induction alone has been reported to provide BLI accuracy close to 0. Nevertheless, iterative self-learning can then amplify this initially weak alignment and improve it dramatically (Artetxe, Labaka, and Agirre, 2018a; Sogaard et al., 2019).

3.2.4 Limits of isometry-based unsupervised embeddings

But the "approximate isometric assumption" might not hold for any pair of monolingual embeddings. Søgaard, Ruder, and Vulić (2018) showed that a specific fully unsupervised method, MUSE (Lample et al., 2018), provides high BLI accuracy only if three conditions are met, otherwise the score is close to zero:

1. **Similar word embedding techniques**: an embedding learned with skip-gram cannot be aligned with one learned with CBOW, however different hyper-parameters like window size still allow alignment.
2. **Same domain**: an embedding trained on Wikipedia can only be aligned with another embedding trained on Wikipedia, but not on one trained on data from the European Parliament or on health-related data.
3. **Similar languages**: unsupervised alignment does not work for all language pairs. For example, MUSE (Lample et al., 2018) fail to align English and Finnish embeddings.

Artetxe, Labaka, and Agirre (2018a) find that, for some pairs of languages, MUSE and another adversarial method (Zhang et al., 2017a) suffer from an instability in their results, where some runs

can lead to accuracy below 5%. For example, MUSE is reported to fail in 4/6 cases for aligning English and Spanish embeddings and 7/10 for English and Italian.

However Artetxe, Labaka, and Agirre (2018a) also introduces a new unsupervised method, VecMap, that is more robust providing systematically good alignment for the pairs it was evaluated on. VecMap introduces a new iterative self-learning that randomly ignore elements from the inferred dictionary with a progressively decreasing probability, which is expected to make the method more robust, in an approach inspired from simulated annealing. This stochastic iterative self-learning has been designated as the reason for the stability of VecMap, both by the authors (Artetxe, Labaka, and Agirre, 2018a) and other works (Sogaard et al., 2019). This raises a new open question, that is also asked by Sogaard et al. (2019):

Open Question 3.2. *Could VecMap self-learning algorithm be used and provide more stable results with other kinds of initialization like adversarial methods?*

Indeed, VecMap, contrary to MUSE and the other compared method (Zhang et al., 2017a), does not rely on adversarial learning for the initialization but rather on a graph-matching inspired method. And, as suggested earlier, adversarial learning suffers from instability. An instability that is inherent to adversarial learning in general, which can suffer from mode collapse (Goodfellow et al., 2014). Therefore, our previous open question leads to another one:

Open Question 3.3. *Is VecMap more robust because of its stochastic iterative self-learning or because of its initialization, or even both?*

Hartmann, Kementchedjhieva, and Søgaard (2019) provide some pieces of the answer to this latter question by showing that Vanilla GANs actually provide better alignment than other more recent seed dictionary induction methods and that the improvement brought by VecMap and others might actually stems from the iterative self-learning.

VecMap can reduce some limitations of unsupervised mapping-based methods without relinquishing the "approximate isometric assumption". Furthermore, many works that have identified such limitations (Artetxe, Labaka, and Agirre, 2018a; Hartmann, Kementchedjhieva, and Søgaard, 2018; Søgaard, Ruder, and Vulić, 2018) only evaluated MUSE, or another adversarial method in the case of Artetxe, Labaka, and Agirre (2018a), leading us to another question:

Open Question 3.4. *Are those three limitations identified by Søgaard, Ruder, and Vulić (2018) due to a lack of approximate isometry between the initial embeddings, or due to the instability of the adversarial approach? In other words, could a more robust isometry-based method like VecMap overcome any of the three conditions?*

For distant and low-resource languages, Vulić et al. (2019) have already performed more extensive experiments and shown that Vecmap does not solve the issue. They show that even supervised baselines provide low BLI accuracy. This thesis also takes an interest in understanding the need for the second condition (the need for embeddings of same domain), which is less investigated.

3.2.5 Measuring the deviation from isometry

Several works have already started to address this question by trying to measure how much a pair of embeddings deviates from isometry. Søgaard, Ruder, and Vulić (2018) who shows the limitations of MUSE, also find a high correlation between BLI accuracy and a graph similarity metric of their design. Patra et al. (2019) show that the approximation of the Gromov-Hausdorff distance provides even higher correlation (or rather anti-correlation, since it's a distance) with BLI accuracy. Since it will be used in this thesis, the computation of the approximation of the GH distance is detailed in Appendix 1 for the interested reader.

Both proposed metrics provide some insight into how much two embeddings deviate from isometry. The graph similarity metric measures how isospectral two neighborhood graphs are. While isospectral graphs are not necessarily isomorphic, isomorphic graphs are always isospectral (Gordon, Webb, and Wolpert, 1992). On the other hand, the Gromov-Hausdorff distance is an exact measure

of the deviation from isometry but it is intractable to compute in practice, so Patra et al. (2019) rely on an approximation based on topological data analysis, which provides only a lower bound on the distance (Chazal et al., 2009).

Eventually, (Marchisio et al., 2022) show that those metrics are not reliable enough for more fine-grained comparisons. Patra et al. (2019) obtained high correlations by comparing runs across languages, with wide variations of BLI accuracy. But when looking at in-language variations, Marchisio et al. (2022) show that the existing metrics become less reliable.

Open Question 3.5. *If isometry metrics correlate well with variations of BLI accuracy across languages, but not across more similar settings, does this mean that the existing metrics are not a precise enough measure of the deviation from isometry or does it mean that something else is affecting alignment?*

3.2.6 Beyond isometry

Because the isometry assumption seems to be too strict of a requirement, several methods have tried to alleviate the need for such a condition. Marchisio et al. (2022) and Patra et al. (2019) have designed approaches that impose a soft isometric constraint. However Patra et al. (2019) is proposing a semi-supervised method using a small training bilingual dictionary, while Marchisio et al. (2022) is the only one to propose a truly unsupervised method that weakens the need for isometry.

Artetxe et al. (2020) question the very need for fully unsupervised alignment methods. They argue that it relies on an unrealistic setting, where no parallel data is available at all, and where sufficient monolingual data is provided to build a monolingual embedding. If both languages are high-resource enough to benefit from large amounts of monolingual data, they must surely be present in some parallel corpora.

However this thesis argues that unsupervised methods are still useful, at least for low-resource languages. As for high-resource ones, unsupervised methods can, in some cases, outperform supervised ones (Lample et al., 2018). Therefore, unsupervised methods remain a worthy object of study, even for languages which can benefit from the availability of large amount of cross-lingual training data.

3.3 Conclusion

Fully unsupervised mapping-based methods seem to be relevant only to very specific settings. Using such methods in the clinical domain would most surely fail, as in-domain monolingual data might not be comparable across language as general-domain data can be. For example, one might rely on biomedical scientific articles to build an English monolingual embedding. But there might not be any non-English equivalent to build an isometric embedding to align it with.

However, while existing unsupervised static embeddings require some strong assumptions, the next chapter shows that multilingual contextualized embeddings can be built even without any explicit cross-lingual signal. Most of them are even pre-trained and fine-tuned without ever specifying in which language the input is written.

Chapter 4

Multilingual Contextualized Embeddings

Multilingual static embeddings, whether unsupervised or not, are always optimized for a formulated cross-lingual objective. The cross-lingual objective can be implicit in pseudo-mixing approaches where artificial code-switching is injected in the pre-training data, but there is always an explicit cross-lingual objective or cross-lingual training signal.

This is not necessarily the case for contextualized embeddings. Multilingual encoding LMs like mBERT (Devlin et al., 2019) and XLM-R (Conneau et al., 2020b) have strong Cross-lingual Transfer Learning (CTL) abilities, despite not having been trained with any explicit cross-lingual objective nor explicit cross-lingual training signal. Instead, they are trained on a collection of monolingual corpora in different languages, with the same training objective as some monolingual LMs. For the sake of readability, those multilingual encoding LMs without any cross-lingual training signal will be said to be "language-agnostic" in the rest of this thesis.

This chapter gives an overview of those mLMs. It also documents attempts at improving these models with a cross-lingual signal. Indeed, while mBERT and XLM-R can show strong CTL performances between similar languages, they struggle with low-resource languages or when transferring between distant ones (Lauscher et al., 2020). Some works have added cross-lingual objectives during pretraining, leveraging translation datasets. Others have tried to improve existing pre-training models by training them on a specific cross-lingual task. Namely, there have been some attempts at enforcing better Multilingual Alignment (MA) of the inner representations of those models. The relative failure of those methods is the main focus of this chapter, as it questions the link between MA and CTL which is studied in our contribution.

4.1 Multilingual Language Models without Cross-lingual Signal

Language agnostics mLMs go against the general principle applied for multilingual static embeddings. Formally, the cross-lingual loss in the Equation 3.1 from the previous chapter is simply removed:

$$J = \sum_{i=1}^{n} \mathcal{L}_i + \cancel{\sum_{i=1}^{n} \sum_{j=1 \neq i}^{n} \Omega_{ij}} \tag{4.1}$$

The rest of the section first describes the mBERT model, which is the first language-agnostic mLM, and then provides a list of such models that is as exhaustive as possible, before briefly commenting on some attempts at explaining their CTL abilities.

4.1.1 mBERT and the absence of explicit cross-lingual signal

mBERT was released simultaneously with BERT (Devlin et al., 2019) and it does not have a dedicated paper[5]. Between BERT[6] and mBERT, only two things change:

- The training data: which becomes multilingual
- The tokenizer: which is augmented in capacity

Regarding the training data, BERT is trained on the English Wikipedia and on a book corpus, while mBERT is trained solely on Wikipedia but in the 100 most frequent languages it contains[7]. A special temperature sampling strategy has been used to oversample rarer languages and undersample more frequent ones. Formally, chunks of texts are sampled in the i-th language with the following probability:

$$p_i = \frac{N_i^\alpha}{\sum_k^L N_k^\alpha} \tag{4.2}$$

The numbers N_k of chunks of text for each of the L languages are exponentiated to α (0.7 for mBERT). The sampling probability of one language i then becomes proportional to this exponentiated N_i^α. This smoothes the distribution of languages. For example, English is sampled 100 times more often than Icelandic, while it has a corpus that is 1,000 times larger.

Regarding the tokenizer, BERT and mBERT both use a WordPiece tokenizer (Wu et al., 2016). This tokenizer decomposes some words into subwords. It has a vocabulary size that is fixed beforehand. To build the vocabulary of subwords, it starts with all possible characters in the vocabulary, then it iteratively merges subwords to create new longer ones that are likely to occur in the training corpus, until it reaches the capacity. The resulting list of subwords provides a deterministic tokenizer that splits rare or OOV words into more frequent subwords. mBERT and BERT use the same tokenizer, but the vocabulary size has been increased for mBERT (from 30k to 110k) to accommodate more languages. Even for languages where additional pre-processing could have been performed, nothing particular was done differently to English. The only particular treatment is for languages that do not use whitespaces like Chinese. In that case, the text is simply tokenized by character.

Besides those differences concerning the training data and the tokenizer capacity, mBERT has the same model architecture as monolingual BERT and the same training objectives[8]. mBERT is thus deprived of any explicit cross-lingual signal and is even been given no explicit information regarding the language in which the input is written, contrary to multilingual static embeddings which, even when aligned in an unsupervised manner, are optimizing for an explicit cross-lingual objective and have separate vocabulary for each language.

Despite its language-agnostic training, mBERT was shown to have surprising Cross-lingual Transfer Learning (CTL) abilities. Pires, Schlinger, and Garrette (2019) showed on an NER and a POS-tagging task that mBERT can be fine-tuned in one language and provide high accuracy when evaluated in another. They also show that this cross-lingual transfer can also work between languages written in different scripts, for which there is no vocabulary overlap between the datasets used for fine-tuning and evaluating[9]. They however show that CTL with mBERT has some limitations. Transfer works better for similar languages. Namely, they find that CTL works better between languages with the same subject/verb/object order or with the same adjective/noun order. They also observe that language pairs with higher vocabulary overlap tend to benefit from better performances with CTL.

[5] Most of the information available can be found on a README on Github https://github.com/google-research/bert/blob/master/multilingual.md

[6] its Base version

[7] The initially released model contains 100 languages including two that have separate writing standards: Chinese (traditional and simplified) and Norwegian (Nynorsk and Bokmål). Mongolian and Thai were added to a new version of the model, which is the version used in most of subsequent works.

[8] Masked Language Modeling (MLM) and Next Sentence Prediction (NSP)

[9] Althoug there might be some overlap in a larger corpus, like the pre-training data

Wu and Dredze (2019) made similar experiments with more tasks and tried to provide some understanding of how CTL works. Across five tasks, they show that mBERT is competitive with strong baselines trained with cross-lingual signal, and is sometimes state-of-the-art. In four of those five tasks, it beats baselines based on cross-lingual static embeddings.

4.1.2 Other mLMs without cross-lingual signal

Following the success of mBERT, several other mLMs were released, with a multilingual pre-training and without explicit cross-lingual signal. They can be differentiated according to three main criteria:

- Model architecture
- training objectives
- training data

4.1.2.1 Architectural differences

Besides models based on BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019b) which uses the same model architecture, only two variants of BERT were used in a multilingual setting without cross-lingual signal:

- RemBERT (Chung et al., 2021) which decouples input and output embeddings,
- DeBERTa (He et al., 2021), which uses positional embeddings differently.

There are also two main alternatives for the tokenizer used. It is either WordPiece as in mBERT or SentencePiece (Kudo and Richardson, 2018), which improves upon WordPiece by treating whitespaces as any other character, so instead of merging subwords back into words, they are merging them back into pieces of sentence that can include whitespaces. This allows to treat all languages in the same manner, whether or not they use whitespaces.

Despite variations in model architecture, all existing encoding mLMs are based on the Transformer encoder architecture. There doesn't seem to be any attempt at building language-agnostic encoding mLMs with recurrent neural networks, maybe because they are less parallelizable and thus harder to train. This raises the following question:

Open Question 4.1. *Does language-agnostic training only lead to cross-lingual abilities for Transformer encoder-only models? What would happen if we performed the same kind of training for learning static word embeddings or generative LMs?*

For generative LMs, the question has been already studied in the literature and there are already multilingual encoder-decoder models like mBART (Liu et al., 2020b) or mT5 (Xue et al., 2021) that are proof of this possibility, but they also rely on the Transformer architecture in its encoder-decoder version. As for other kinds of encoding LMs, the question remains, namely for static word embeddings.

4.1.2.2 Training objectives

Most language-agnostic mLMs use Masked Language Modeling (MLM). While mBERT also uses Next Sentence Prediction (NSP), it is dropped in many mLMs based on the observation that it can be detrimental to downstream tasks (Liu et al., 2019b).

MLM can also be replaced by ELECTRA (Clark et al., 2020) which is known to be more efficient in terms of training samples needed to reach a given downstream performance. ELECTRA relies on a smaller language model (the generator), which is trained to perform MLM on a given partially-masked input and a larger language model (the discriminator) trained on the output of the smaller LM to distinguish between original tokens and inferred ones, as shown on Figure 4.1. The end product is the discriminator, while the generator is discarded.

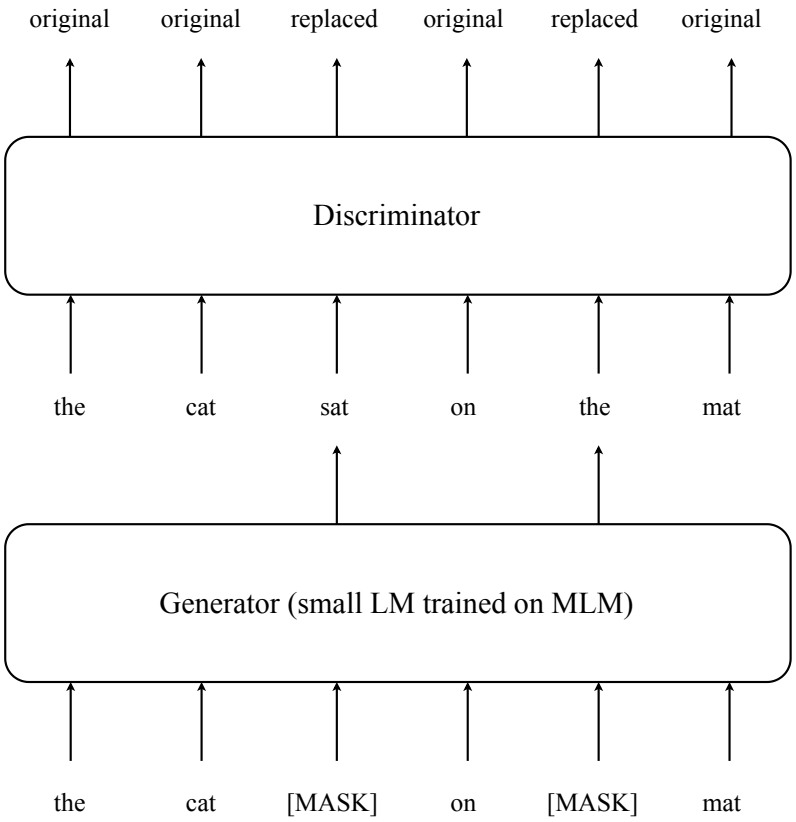


Figure 4.1 – Schema of the ELCTRA pre-training procedure. A generator is trained to perform MLM on a given masked text and a discriminator is jointly trained to predict which words were originally masked. The training is joint but not adversarial, the generator has its own pre-training objective and is not trained to fool the discriminator.

ELECTRA is used as is in some encoding mLMs like SERENGETI-E110 and SERENGETI-E250 (Adebara et al., 2023), and it can be tweaked. This is the case of mDeBERTav3 (He, Gao, and Chen, 2023) which builds upon DeBERTa and ELECTRA. In the original ELECTRA, the first embedding layer of the generator and discriminator is shared, and its weights are updated according to backpropagation through both models, while in DeBERTav3, it is still shared, but only updated according to the gradients backpropagated through the generator, not to those of the discriminator.

4.1.2.3 Training data

The different existing language-agnostic mLMs are all trained on large corpora of text written in multiple languages. Since there is no explicit cross-lingual signal, the whole training data can be seen as a collection of monolingual corpora.

The pre-training data of language-agnostic mLMs differ according to two main perspectives:

- The group of languages selected
- The pre-training corpus, and the different domains it covers

Selection of pre-training languages The most-studied language-agnostic mLMs are pre-trained on the most frequent languages found in the original pre-training corpus. mBERT is trained on the 102 most important languages of Wikipedia. XLM-R (Conneau et al., 2020b) is trained on the 100

most present languages of the Common Crawl corpus. Other models are dedicated to specific groups of languages, usually based on geographic proximity, such as Indian languages with IndicBERT (Kakwani et al., 2020) or African ones with SERENGETI (Adebara et al., 2023), AfriBERTa (Ogueji, Zhu, and Lin, 2021), and AfroLM-LARGE (Dossou et al., 2022).

It is worth noting that geographic proximity between languages does not necessarily mean that those languages will necessarily share a lot of common features. It could be argued that the pre-training languages for IndicBERT or AfriBERTa are not necessarily less diverse than for mBERT or XLM-R, which can be seen as Western-centric language models.

Language-agnostic mLMs are typically trained in dozens or hundreds of languages. To date, glot500-m (ImaniGooghari et al., 2023), trained in 500 languages, is the language-agnostic model that is trained with the highest number of languages. However, there might not be any definitive answer about whether using more languages in pre-training improve overall results. When releasing XLM-R, Conneau et al. (2020b) showed that there could be a "curse of multilinguality" where pre-training in too many languages might lead to interferences between languages and reduce overall CTL abilities. They also show that increasing the model size only partly solves the issue. On the other hand, Dufter and Schütze (2020) show in a small controlled setting that reducing the capacity of the model might force it to share representations between languages, making it more able to perform Cross-lingual Transfer Learning (CTL). Finally, when releasing glot-500-m, ImaniGooghari et al. (2023) show that having many languages in the pre-training dataset is beneficial to most languages, especially low-resource ones, with the exception of language isolates[10], suggesting that pre-training in many languages helps the model rely on related languages for generalization to a given language.

Quantity and quality of the pre-training data All existing Language-Agnostic Language Models (LALMs) are pre-trained on general-domain textual data. Most of them have focused on the quantity of pre-training data rather than its quality. mBERT started with the Wikipedia corpus, but the XLM-R authors showed that using Common Crawl, a larger corpus scrapped from the web provided a better coverage of low-resource languages but also better downstream results on all languages, high-resource ones included.

The reader might wonder whether the corpora need to be comparable across languages. Indeed, in the case of Wikipedia, a lot of documents can be the translation of others, and even if they are not, there are many Wikipedia articles covering the same subjects. Moreover, the previous chapter has shown that using corpora of different domains for different languages made unsupervised methods for cross-lingual static embeddings fail. However, this does not seem to be the case for contextualized embeddings. Conneau et al. (2020a) show that it is possible to build bilingual language-agnostic LMs with pre-training data from different domains.

Some more recent works have tried to improve upon the quality of the pre-training dataset. For example, glot-500-m authors (ImaniGooghari et al., 2023) propose the glot-500-c corpus, where they apply several filters to eliminate samples that have too much character or token repetition, that are duplicates of another, that contain too few words, or for which the language seems to be misidentified, but it can also attempt to remove toxic content from the pre-training data. However, there is no controlled experiment about the impact of such filters on downstream multilingual abilities.

4.1.3 Re-using existing language models

Since pre-training is expensive, it is worth noting that several works do not start from randomly initialized models but rather from existing language-agnostic mLMs like mBERT and XLM-R. Glot-500-m, which is already cited extensively throughout this chapter, is actually an extension of XLM-R Base to up to 511 languages.

To extend an existing language-agnostic mLM, existing works usually continue the pre-training on the same training objective as the original model: Masked Language Modeling (MLM). But it

[10] languages that are not closely similar to any other language like Basque.

is also needed to update the vocabulary to accommodate the additional languages. For example, Glot-500-m authors train a new SentencePiece tokenizer on the pre-training data to obtain a new vocabulary of subwords. They then merge this vocabulary with XLM-R vocabulary which adds 151k new tokens.

Language-agnostic mLMs have also been reused to be compressed into smaller models. The knowledge distillation proposed by Sanh et al. (2019) was used to create distilmBERT from mBERT and another compression technique was also used to obtain XLM-R-MiniLM from XLM-R (Wang et al., 2020).

Re-training a language-agnostic mLM can also allow one to inject a cross-lingual training signal to the model as will be detailed later in the chapter.

4.2 Attemps at explaining CTL abilities

A whole line of work is trying to explain how CTL abilities can emerge from mLMs despite the absence of an explicit cross-lingual signal. Philippy, Guo, and Haddadan (2023) provides the most recent review focused on this subject. They list five factors that have been shown to affect CTL:

- linguistic similarity between languages
- the amount of shared vocabulary: it was shown to correlate with CTL abilities (Pires, Schlinger, and Garrette, 2019) but this link was disputed by other works (Conneau et al., 2020a)
- model architecture: deeper models work better (K et al., 2020) but smaller hidden size might also be needed (Dufter and Schütze, 2020). It was also shown that sharing parameters across languages in the upper layers was crucial (Conneau et al., 2020a)
- the pre-training setting: the fertility of a tokenizer for a given language has a negative impact on the transfer to that language (Rust et al., 2021), and the Next Sentence Prediction objective of mBERT is actually detrimental to CTL (K et al., 2020)
- the pre-training data: The size of the pre-training data seems to correlate with the CTL abilities of a given model (Ahuja et al., 2022; Lauscher et al., 2020; Liu et al., 2020a; Srinivasan et al., 2021) and domain mis-matches between the training data of each language decrease cross-lingual transfer abilities (Conneau et al., 2020a).

Eventually, the reasons for the cross-lingual abilities of mBERT and other models are still the object of ongoing debates in the literature. But these investigations are not the focus of this thesis, although its contribution on the link between multilingual alignment and cross-lingual transfer can shed some light on the understanding of those models.

4.3 Adding cross-lingual signal

Language-agnostic models like mBERT or XLM-R show cross-lingual abilities without explicit cross-lingual training. This ability could be improved and some works have attempted to inject an explicit cross-lingual signal during the pretraining of similar models or on an already pretrained model.

4.3.1 During pre-training

Several cross-lingual pre-training objectives can be devised for encoding LMs. Most of them rely on the same kind of cross-lingual signal: parallel data. During pre-training, the model is fed pairs of translated sentences and is optimized for a given training objective. The simplest of such training objectives is Translation Language Modeling (TLM) (Conneau and Lample, 2019), which is simply MLM but where the input is the concatenation of two translated sentences. The model is then

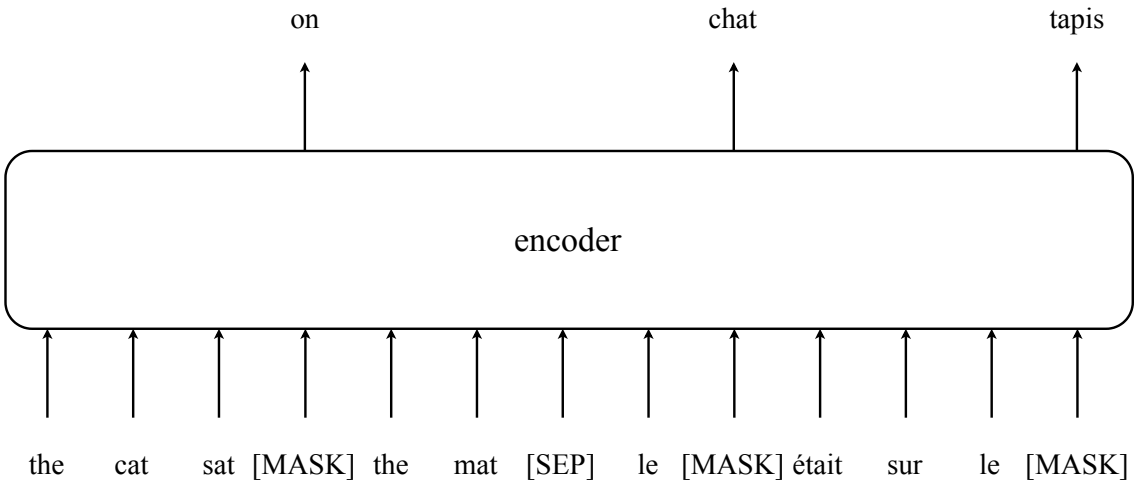


Figure 4.2 – Translation Language Modeling (TLM): a pair of translated sentences is provided as input to the model and random words are masked out. By trying to recover the masked words, the model is expected to rely on the translation.

expected to rely on the translation as well as the context to find the masked-out token, as shown in Figure 4.2.

There are many other cross-lingual pre-training objectives in the literature, but we let the reader refer to Doddapaneni et al. (2021) for an exhaustive inventory. Our focus is on encoding mLMs that are language-agnostic during pre-training. Nevertheless, this thesis takes a specific interest in the injection of a cross-lingual signal after pre-training, particularly in methods that try to improve the multilingual alignment of contextualized embeddings, because the relative failure of those mehods (Wu and Dredze, 2020) questions the very link between multilingual alignment and CTL.

Besides changing the pre-training objectives, Yang et al. (2020) add cross-lingual signal directly in the pre-training data by artificially mixing languages in input sentences, similarly to what was done with pseudo-mixing methods for static embeddings (cf. Section 3.1.2).

4.3.2 After pre-training

Assuming that better-aligned contextualized embeddings lead to better cross-lingual transfer, several works have proposed methods for aligning multilingual contextualized representations produced by models like mBERT and XLM-R to improve their CTL abilities. This section focuses on those supervised alignment methods that are similar to realignment methods in cross-lingual static embeddings. It then briefly enumerates other methods that inject a cross-lingual signal after pre-training.

4.3.2.1 Supervised realignment

Realignment methods, sometimes called adjustment or explicit alignment, aim to improve the cross-lingual properties of an mLM by trying to make similar words from different languages have closer representations. They typically require a translation dataset and an alignment tool, like FastAlign (Dyer, Chahuneau, and Smith, 2013), to extract contextualized pairs of translated words that will be realigned, as detailed in Section 2.3.4. Then, those pairs of contextualized translated words can be used as a training signal for realignment, as is a bilingual dictionary for aligning static embeddings.

One of the first attempts at supervised realignment of contextualized representations was actually directly inspired by the alignment of static embeddings. Wang et al. (2019), in the exact same

manner as supervised BLI, use Procrustes to learn an orthogonal mapping between pairs of translated words. The details of the method were already described in the previous chapter (cf. Equation 3.4). The only difference is that the representations that are aligned are not uncontextualized entries of a bilingual dictionary, but rather contextualized representations of words extracted from translated sentences with FastAlign.

While the previous method does not update the weights of the model, other methods rely on training the whole model to produce better-aligned representation instead of learning an additional orthogonal mapping to apply to the representation. Cao, Kitaev, and Klein (2020) optimize for the same objective as previous work but train the whole mBERT model. Their $\ell 2$ loss is written as:

$$L_{\ell 2}(\theta) = \sum_{i=1}^{N} ||f(s_i;\theta) - f(t_i;\theta)||_2^2 \tag{4.3}$$

Where f is a function representing the model with learnable parameters θ, s_i and t_i are a pair of input translated words (among N training pairs), given with their respective context to the model.

Because Cao, Kitaev, and Klein (2020) are training the whole model instead of an orthogonal matrix, they need to add a regularization term to avoid the degenerative solution of f returning a constant. They add to the loss a regularization term that encourages the current representations of the target language to be similar to its initial representation before any realignment, which gives, with θ_0 being the model parameters before any realignment:

$$R(\theta) = \sum_{i=1}^{N} ||f(t_i;\theta) - f(t_i;\theta_0)||_2^2 \tag{4.4}$$

Zhao et al. (2021) use the same loss $L_{\ell 2} + R$ but they add a a language-specific layer of batch normalization, which centers and standardizes the representations of each language at the last layer.

If the previously mentioned methods need to rely on a regularization term, it is mainly because they are minimizing the distances between similar pairs without taking dissimilar pairs into account. In other words, they are maximizing the similarity of positive pairs but are not minimizing the similarity of negative pairs. Adding the regularization term helps but other works have used a contrastive loss instead of the $\ell 2$ loss, and thus take into account positive and negative pairs in a single loss term.

Wu and Dredze (2020) proposes such a contrastive loss, inspired by a computer vision framework (Chen et al., 2020). The similarity between the representations of a positive pair is maximized with respect to all in-batch negative pairs involving one element of the positive pair. The loss for a given batch is written as follows:

$$\begin{aligned} L_{\text{weak}}(\theta) = \frac{1}{2B} \sum_{i=1}^{B} \Bigg(& \log \frac{\exp\left(\frac{\cos(f(s_i;\theta), f(t_i;\theta))}{T}\right)}{\sum_{j=1}^{B} \exp\left(\frac{\cos(f(s_i;\theta), f(t_j;\theta))}{T}\right)} \\ & + \log \frac{\exp\left(\frac{\cos(f(s_i;\theta), f(t_i;\theta))}{T}\right)}{\sum_{j=1}^{B} \exp\left(\frac{\cos(f(s_j;\theta), f(t_i;\theta))}{T}\right)} \Bigg) \end{aligned} \tag{4.5}$$

cos is the cosine similarity. T is a fixed hyperparameter called temperature. The whole loss boils down to a sum of contrastive terms of the following form:

$$\text{contr}_T(s, t, \mathcal{H}; \theta) = \log \frac{\exp\left(\frac{\cos(f(s;\theta), f(t;\theta))}{T}\right)}{\sum_{s',t' \in \mathcal{H}} \exp\left(\frac{\cos(f(s';\theta), f(t';\theta))}{T}\right)} \tag{4.6}$$

In the above equation, (s, t) is a positive pair and $\mathcal{H}$ is a set containing the positive pair and negative pairs. In the case of L_{weak}, the set of negative and positive pairs only contains cross-lingual

pairs, i.e. pairs of words involving one word from the source language and one from the target language. This enforces weak alignment. Wu and Dredze (2020) also propose a strong alignment loss, where positive pairs are enforced to be more similar than any other cross-lingual negative pairs, but also than any pair involving distinct words from the same language, as per the definition of strong alignment detailled in Section 2.3.2.

L_{weak} eventually rewrites as:

$$L_{\text{weak}} = \frac{1}{2B} \sum_{i=1}^{B} \text{contr}_T(s_i, t_i, \mathcal{H}_{i,\text{left}}) + \text{contr}_T(s_i, t_i, \mathcal{H}_{i,\text{right}}) \tag{4.7}$$

With $\mathcal{H}_{i,\text{left}} = \{(s_i, t_1), ..., (s_i, t_B)\}$ and $\mathcal{H}_{i,\text{right}} = \{(s_1, t_i), ..., (s_B, t_i)\}$. The strong alignment loss, L_{strong}, simply replaces the sets $\mathcal{H}_{i,\text{left}}$ and $\mathcal{H}_{i,\text{right}}$:

$$L_{\text{strong}} = \frac{1}{2B} \sum_{i=1}^{B} \text{contr}_T(s_i, t_i, \tilde{\mathcal{H}}_{i,\text{left}}) + \text{contr}_T(s_i, t_i, \tilde{\mathcal{H}}_{i,\text{right}}) \tag{4.8}$$

The new sets of contrastive pairs are defined as follows and includes same-language pairs as negative samples for the contrastive terms:

$$\tilde{\mathcal{H}}_{i,\text{left}} = \mathcal{H}_{i,\text{left}} \cup \{(s_i, s_1), ..., (s_i, s_B)\} \backslash (s_i, s_i) \tag{4.9}$$

$$\tilde{\mathcal{H}}_{i,\text{right}} = \mathcal{H}_{i,\text{right}} \cup \{(t_1, t_i), ..., (t_B, t_i)\} \backslash (t_i, t_i) \tag{4.10}$$

It of course excludes (s_i, s_i) and (t_i, t_i), the similarity of which will always be 1 which cannot be considered to be relevant negative pairs.

4.3.2.2 The relative failure of realignment methods

When introduced, the mapping-based realignment method of Wang et al. (2019) demonstrated some improvement in a dependency parsing task. Similarly, the $\ell 2$ loss from Cao, Kitaev, and Klein (2020) and Zhao et al. (2021) showed improvements on NLI. However, those improvements might not hold across tasks or language pairs. Kulshreshtha, Redondo Garcia, and Chang (2020) show, in a comparative study, that mapping-based realignment is only effective on "moderately close languages", whereas $\ell 2$ loss with regularization improves results for "extremely distant languages". Efimov et al. (2023) show that this $\ell 2$ loss with regularization works well on NLI but not for all languages on a NER task, and is even detrimental to cross-lingual question answering. Most importantly, Wu and Dredze (2020), who introduced the contrastive learning losses, perform the comparison of all the realignment methods evoked above and show that when compared across several random seeds, tasks, and languages, there does not seem to be any realignment method that consistently outperforms the baseline.

Realignment methods increase the multilingual alignment of the inner representations of a model. But they do not necessarily improve the downstream CTL abilities. This raises the following question:

Open Question 4.2. *Is multilingual alignment correlated with cross-lingual transfer?*

The contribution of this thesis will show in later chapters that multilingual alignment is indeed correlated with CTL abilities. This leads to an additional question:

Open Question 4.3. *What are the conditions for the success of realignment in terms of downstream cross-lingual performances?*

4.3.2.3 Other ways to inject cross-lingual signal after pre-training

The relative failure of realignment methods is all the more surprising that other ways to inject a cross-lingual signal after pre-training have been successful. But those methods have not been necessarily designed to encourage the alignment of representations, but rather fit in either of the three following framework:

- post-training: where the pre-training is continued with pre-training objectives, preferably with explicitly cross-lingual ones
- post-processing: where a transformation is applied to the produced representation with the goal of suppressing language-specific features
- Parameter-Efficient Fine-Tuning (PEFT): where, like in post-training, the pre-training is continued, but instead of updating the whole model weights, it learns additional transformations that are applied at each layer, similar to post-processing.

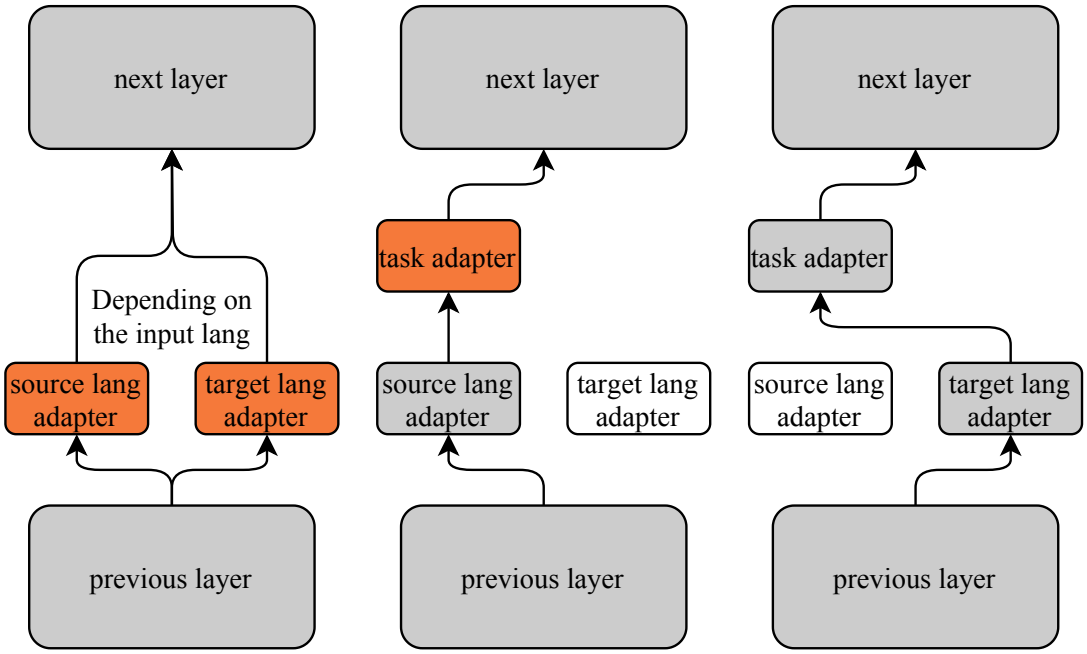
Post-training In this framework, the pre-trained weights are updated according to a new training objective, before using the model for fine-tuning on any task. All realignment methods evoked above except the mapping-based one are specific post-training methods. However, other methods that are not necessarily "alignment-oriented" have been designed. Pan et al. (2021) continue training with a sentence-level contrastive loss that enforces sentence-level multilingual alignment, but at the word level, it uses TLM which does not necessarily enforce alignment. Kondratyuk and Straka (2019) perform fine-tuning on multiple languages for POS-tagging and other tasks derived from syntactic parsing to improve results on syntactic parsing, including for languages not seen during training.

Instead of improving multilingual alignment, it was proposed to adversarially remove language-specific components of the representations. Libovický, Rosa, and Fraser (2020) create `lng-free` by training mBERT adversarially to fool a classifier tasked to determine the language of the input. This method is called adversarial language de-identification. While they do not measure the impact of such a method on CTL abilities, Tanti et al. (2021) use a similar method and do not find that it increases CTL abilities.

The failure of adversarial language de-identification echoes the flaws of realignment methods. However, the lack of success of the former might be due to other reasons. It is inspired by the adversarial removal of demographic attributes proposed by Elazar and Goldberg (2018). But this latter work shows that, in the case of demographic information, the method is not fully reliable. Namely, after adversarial training, a new classifier trained on unseen data might still perform well. It could also be the case for adversarial language de-identification.

Post-processing In this framework, the weights of the model are left untouched but the representations it produces are post-processed in some language-specific manner. For example, the mapping-base realignment of Wang et al. (2019) learns a mapping to apply to each non-pivot language to map them to the pivot language (English). It was also shown that centering representations by removing language centroids could increase sentence retrieval results (Libovický, Rosa, and Fraser, 2020; Pires, Schlinger, and Garrette, 2019). As for evaluating the impact of such centering on CTL, Zhao et al. (2021) are performing some centering additionally to their realignment method. They perform an ablation analysis and thus show that the normalization alone, which performs some scaling additionally to centering, sightly improves CTL results. But this improvement is small with respect to the one the measure for the complete realignment method which was however shown to not provide consistent improvement across tasks and languages by Wu and Dredze (2020).

Parameter-Efficient Fine-Tuning (PEFT) This last framework can be seen as a solution in-between post-training and post-processing. The existing PEFT methods for multilingual language models rely on adapters. Initially designed for cross-domain representations in computer vision (Rebuffi, Bilen, and Vedaldi, 2017), they were eventually brought to NLP (Houlsby et al., 2019). Adapters are small learnable subnetworks that are added to existing models, usually in between



(a) Masked Language Modeling on source and target languages

(b) Fine-tuning on source language

(c) Evaluation on target language

Figure 4.3 – Oversimplification of the MAD-X adapter fine-tuning. This ignores skip-connections and layer normalization in which adapters are actually intertwined. It also doesn't show additional inverse adapters that are used at the input and output of the model. This schema simply explains how each language-specific and task-specific set of adapters is trained. Orange indicates that the weights are updated during the given step. First, language-specific adapters are trained for Masked Language Modeling (MLM). Then, the task adapter is stacked on the source language adapter for fine-tuning for the downstream task in the source language. Finally, the task adapter is stacked on the target language adapter for evaluation on that same task in the target language.

layers. However, the use of a given adapter depends on the domain of the input. There are several ways to use adapters, but a typical one is to have one set of adapters for each domain and to perform the forward and backward pass in the model only through the adapters corresponding to the domain of the input. But as there are domain-specific adapters, there can also be task-specific adapters, or in the case of multilingual NLP: language-specific adapters. This thesis will not delve into the details of Transformers adapters, but will simply describe the basic principle of an adapter scheme used to improve mLMs: MAD-X (Pfeiffer et al., 2020). MAD-X, schematized in Figure 4.3, uses a language-specific adapter for each language. They are first trained with MLM with the whole model, with the rest of the model remaining frozen. Then, language adapters are also frozen and a task-specific adapter is stacked on the source language adapter and trained on the fine-tuning task in the source language. Finally, evaluation can be performed by freezing and stacking the task-specific adapter onto the target language adapter.

Pfeiffer et al. (2020) report that MAD-X provides significant improvements on an NER task for many language pairs, although they show a detrimental impact for some low-resource languages. It is unclear whether MAD-X provide more consistent improvements across tasks and languages than realignment methods.

4.4 Conclusion

Contrary to static multilingual embeddings, contextualized multilingual embeddings can be built without any cross-lingual training signal. Some of them are not even given any indication about the language of the input. Despite that, they show surprising cross-lingual transfer abilities and outperform baselines based on static embeddings.

However, the reasons for those transfer abilities are still to be clearly identified, although some important factors have already been established. Providing a better understanding of those language-agnostic contextualized embeddings could help improve cross-lingual transfer abilities.

This thesis participates in this pursuit of a better understanding of mLMs by focusing on the link between Multilingual Alignment (MA) and Cross-lingual Transfer Learning (CTL) in contextualized embeddings. It namely looks at the word-level alignment, where the evaluation for contextualized embedding is non-trivial.

The next chapter briefly compares static and contextualized multilingual embeddings and arrange them in a broader typology of cross-lingual language models to provide a precise framing of the contribution of this thesis.

Chapter 5

Unanswered questions about multilingual representations

Previous chapters detail the existing extensions of static and contextualized embeddings to a multilingual setting, with a focus on unsupervised ways to obtain such representations. While static multilingual embeddings rely on cross-lingual objectives, contextualized ones can be built without any information on the language of the input embedding. This language-agnostic training for contextualized embeddings can be seen as an absence of cross-lingual supervision that goes even further than fully unsupervised mapping-based methods for multilingual static embeddings.

This chapter puts into perspective the different kinds of multilingual embeddings, defined by different levels of supervision and a distinction between static and contextualized representations. This thesis proposes a typology of multilingual language models, which contains multilingual embeddings but also generative LMs. This typology reveals some gaps in the literature and allows to frame more precisely the scope of this thesis.

This chapter can serve as an overview of the state of the art in multilingual representations but also as an outline for this thesis, focusing on the gaps of the literature this thesis intends to address, and gathering and organizing the open questions that arose in the previous chapters.

5.1 A typology of Multilingual Models

The proposed typology differentiates multilingual representations and language models according to two main characteristics:

- The level of cross-lingual supervision: whether the model relies on parallel data or not, with some additional nuance concerning the type of training signal or the language-agnosticism of the model
- The type language model or language representation, or more precisely what kind of output they produce, whether it is an embedding (static or contextualized) or a generative language model that produces text.

5.1.1 Different levels of cross-lingual supervision

Monolingual language models are usually trained with unsupervised - or rather self-supervised - training objective, like Masked Language Modeling (MLM). For multilingual LMs, the same monolingual objectives can be used, and, as seen in previous chapters, cross-lingual objective can be added, as shown in the equation 3.1 of Chapter 2, that can be schematized as follows:

$$J = \mathcal{L} + \Omega$$

$\mathcal{L}$ represents the monolingual pre-training objective, which can usually be seen as a sum of a monolingual objective for each involved language. Ω is the cross-lingual pre-training objective, which aims at providing some level of multilingual alignment of cross-lingual transfer abilities. This cross-lingual pre-training term can be supervised or unsupervised.

If the cross-lingual objective is unsupervised, two cases can be differentiated:

- Language-aware: when the cross-lingual term is explicit, as in fully unsupervised mapping-based methods for multilingual static word embeddings (cf. Section 3.2.3)
- Language-agnostic: when no explicit cross-lingual term is used, nor any explicit cross-lingual signal is used as would be done with pseudo-code switching or Translation Language Modeling (TLM). mBERT or XLM-R are examples of language-agnostic language models (cf. Chapter 4). In the previous equation, the cross-lingual term Ω would be removed.

The reader must note that any language-specific component of a model will make it language-aware. For example, MAD-X uses language-specific adapters, hence it is deemed language-aware. Models like mBART use a specific language token at the beginning of the input and can thus be considered to be language-aware too.

If the cross-lingual objective is supervised, one can define two different levels of supervision:

- sentence-level supervision: where only a parallel corpus is needed, and the similarity between sentence representations is explicitly enforced.
- word-level supervision: where the training signal is a bilingual dictionary.

Word-level alignment supervision signal might be scarce for contextualized embeddings where pairs of words must be contextualized, i.e. word pairs must be extracted from sentence pairs as explained in Section 2.3.4.

Other levels of cross-lingual supervision can be defined. In their book, Sogaard et al. (2019) organize existing cross-lingual static embeddings according to three levels: word-level, sentence-level, but also document-level. Because this thesis mainly focuses on unsupervised methods and because document-level supervision was only applied to static embeddings, this thesis considers a more flexible definition of sentence-level supervision which includes document-level alignment.

5.1.2 Different sorts of language models and representations

Different types of language models have been applied to a multilingual setting. This study focuses on word embedding produced by shallow neural networks like static embeddings or by encoder-only Transformers for contextualized embeddings. However, the proposed typology should also include generative language models, differentiating two kinds of models, independently of whether they are Transformer-based or not:

- text-to-text models: which include encoder-decoder Transformers and takes an input sequence and predict an output one
- auto-regressive models: which include decoder-only Transformers and generate text from scratch, although they can be used in a text-to-text manner by guiding the generation.

5.1.3 The proposed typology

The proposed typology is shown in Table 5.1 with some examples. It reveals some gaps in the literature.

The most obviously empty part of the typology is for supervised methods on generative models. While training a translation model can be seen as training a text-to-text language model with sentence-level cross-lingual signal, it does not fit well with the proposed typology since translation models do have downstream applications like CTL other than translation itself. Besides translation

	Embeddings		Generative	
	static	contextualized (encoder-only)	text-to-text (encoder-decoder)	auto-regressive (decoder-only)
Unsupervised				
language-agnostic		- Encoder-only Transformers (mBERT, XLM-R)	- encoder-decoder trained with denoising objective (mT5)	- decoder-only trained on text generation (XGLM, Bloom) - multilingual multi-task fine-tuning (BloomZ, mT0)
language-aware	- mapping-based methods - adversarial, ICP (MUSE) - graph-matching (RCSLS, VecMap) - joint alignment (IsoVec)	- language de-identification (lng-free) - adapters (MAD-X)	- language-specific special token (mBART) - adapters (mmT5)	
Supervised				
sentences	sentence-level alignment methods[1]	- TLM - sentence-level realignment	- translation models	
words	- mapping-based (Procrustes) - pseudo-mixing - joint alignment	- realignment		

Table 5.1 – Proposed typology of multilingual models with examples. **Orange** cells show the frontiers in the literature this thesis explores.

[1] This table does not do justice to the diversity of sentence-level alignment methods for static embeddings, as they are not the focus of this thesis. The reader is referred to Chapter 6 of Sogaard et al. (2019) for more details.

models, to the best of my knowledge, there isn't any attempt at improving alignment in generative models or pre-training on explicitly cross-lingual objectives.

Applying re-alignment methods, or using cross-lingual pre-training objectives for generative models might be an interesting venue for research, outside of the scope of this thesis. But it must also be noted that the cross-lingual abilities of multilingual generative models have been improved in other manners. Language-specific adapters were used in encoder-decoder models like mmT5 (Pfeiffer et al., 2023). More importantly, the literature for multilingual generative models has followed the same paradigm as for monolingual models: performing pretraining on a large unlabelled corpus, then fine-tuning or performing more complex learning schemes on a small set of prompt examples (Ouyang et al., 2022; Zhou et al., 2023). This was done in a multilingual setting with multilingual prompts, but not cross-lingual ones, with models like BloomZ and mT0 (Muennighoff et al., 2023).

The proposed typology reveals another gap in the literature that is more relevant to the content of this thesis: there does not exist any language-agnostic static embedding. Since language-aware unsupervised methods already struggle in many settings, it could be anticipated that any attempt at language-agnostic static embedding might not provide a well-aligned embedding. However, this thesis aims at revisiting multilingual static embeddings with the new language-agnostic perspective brought by larger models. It does not provide new state-of-the-art embeddings with high BLI accuracy, but it shines a new light on the failure of existing unsupervised alignment methods and on the link between alignment and pre-training.

While language-agnostic static embeddings are an open venue of research, there is another one at the other end of multilingual embeddings: word-level cross-lingual supervision for contextualized embeddings. While word-level realignment methods do exist, they are not yet completely successful. This thesis investigates the failure of those realignment methods.

5.2 Framing the contribution

In the previous section, two main venues of research for this thesis were laid out: (1) revisiting unsupervised multilingual static embeddings, and (2) investigating multilingual alignment of

contextualized representations. This section details those two research axis and gathers the relevant open questions that arose from the state of the art.

5.2.1 Revisiting unsupervised static embeddings

Knowing that multilingual contextualized embeddings provide good results without any cross-lingual training signal, it underlines a gap identified in the previously proposed typology: what about language-agnostic static embeddings? But the success of language-agnostic contextualized embeddings raises another question: why do unsupervised mapping-based methods for static embeddings fail under certain settings? Are they doomed to fail?

5.2.1.1 A gap in the typology: language-agnostic static embeddings

The contrast between the success of language-agnostic Transformers LMs and the limitations of cross-lingual static embeddings raises the following question[11]:

Open Question 4.1. *Does language-agnostic training only lead to cross-lingual abilities for Transformer encoder-only models? What would happen if we performed the same kind of training for learning static word embeddings or generative LMs?*

This thesis investigates the possibility of language-agnostic static embeddings. It shows that while a vanilla language-agnostic training for word embedding provides near-zero BLI accuracy, it still shows some very weak levels of alignment that is however significantly above a random baseline. This thesis also investigates the causes of the emergence of this alignment and demonstrates the impact of code-switching.

5.2.1.2 Investigating the failure of unsupervised multilingual static embeddings

Having shown that code-switching improves alignment in language-agnostic embeddings, it allows us to propose an answer to the following question:

Open Question 3.1. *While there are unsupervised mapping-based approaches, Is it possible to design pseudo-mixing or explicitly joint approaches in an unsupervised setting?*

This thesis shows that unsupervised alignment methods can rely on code-switching that is already present in the data, instead of using pseudo-mixing, which introduces code-switching in a supervised manner.

Relying on code-switching helps particularly for pairs of languages that are written in different scripts, which are often failure cases of existing unsupervised methods. This leads to investigating the failure of existing unsupervised mapping-based methods.

Namely, when comparing the proposed method based on code-switching with other ones, this thesis also compares different initialization techniques, providing some insights about the two following questions:

Open Question 3.2. *Could VecMap self-learning algorithm be used and provide more stable results with other kinds of initialization like adversarial methods?*

Open Question 3.3. *Is VecMap more robust because of its stochastic iterative self-learning or because of its initialization, or even both?*

Finally, some early work of this thesis investigates the supposed lack of isometry in a cross-lingual setting, with the aim of building multilingual embedding for the clinical domain. Providing some hints of an answer to the last question on static embeddings:

Open Question 3.4. *Are those three limitations identified by Søgaard, Ruder, and Vulić (2018) due to a lack of approximate isometry between the initial embeddings, or due to the instability of the adversarial approach? In other words, could a more robust isometry-based method like VecMap overcome any of the three conditions?*

[11] The open questions in this chapter are copies of important open questions raises throughout the previous chapters, hence the numbering indicates the chapter where they come from.

5.2.2 Investigating multilingual alignment in contextualized embeddings

While unsupervised alignment is an open venue of research for static embeddings, supervised alignment raises many questions for contextualized ones. The failure of realignment methods can make the reader wonder whether a better alignment of contextualized embedding is truly useful. This leads this thesis to pursue two research themes: (1) proposing evaluation methods for contextualized embeddings in order to study the link between alignment and cross-lingual transfer, and (2) investigate the failure of realignment methods.

5.2.2.1 Gaps in the evaluation of multilingual alignment

The review of the literature of this thesis shows that there is a lack of consensus about whether existing contextualized embeddings are well-aligned or not, raising questions about evaluation methods at the sentence and word level:

Open Question 2.2. *How good is the Multilingual Alignment (MA) of raw sentence representations obtained from contextualized word embeddings? What pooling method provides the best alignment?*

This thesis compares different methods for measuring sentence alignment and shows that relying on mean pooling might make more sense than other sentence-level representations. But when looking at word-level alignment, another question arises:

Open Question 2.4. *What is the impact of word alignment errors on the measure of multilingual alignment? How can we mitigate the impact of those errors?*

This thesis shows that word alignment errors are indeed impactful, and proposes a new method for extracting contextualized pairs of translated words for the evaluation of multilingual alignment.
This proposed method can also be adapted to answer the two following questions:

Open Question 2.3. *How do contextualized and static multilingual embeddings compare in terms of word-level multilingual alignment?*

Open Question 2.1. *Are multilingual static embeddings competitive in terms of strong alignment?*

The proposed evaluation methods show that multilingual contextualized embeddings have a multilingual alignment that is comparable with a competitive baseline of static embeddings aligned with a supervised method. However, this comparison only holds for weak alignment, as this thesis shows that static multilingual embeddings are lagging behind contextualized ones in terms of strong alignment.

5.2.2.2 Investigating the failure of realignment methods

Having found that realignment methods fail to consistently improve cross-lingual transfer results, it raises the following question:

Open Question 4.2. *Is multilingual alignment correlated with cross-lingual transfer?*

This thesis shows that there is a high correlation between multilingual alignment and cross-lingual transfer abilities, particularly with strong alignment, which eventually leads to the question:

Open Question 4.3. *What are the conditions for the success of realignment in terms of downstream cross-lingual performances?*

This thesis identifies several factors explaining the failure of realignment.

5.2.3 Application to the clinical domain

While this thesis studies multilingual embeddings from a general perspective, it ultimately aims at applying them to the clinical domain.

For static embeddings, studying the limitations of unsupervised cross-lingual embeddings allows us to show that they are not doomed to fail in a cross-domain setting involving the biomedical domain. But static embeddings are eventually too limited with respect to contextualized models, even from the perspective of multilingual alignment.

Therefore, the last chapter of this thesis focuses on the application of multilingual contextualized embeddings to the clinical domain, in comparison with translation-based methods. Yarmohammadi et al. (2021) have already shown, in the general domain, that translation-based methods can sometimes outperform cross-lingual models, but that it depends on the task, and language pair at hand. Ultimately, it seems that, in the general domain, neither translation-based methods nor CTL works systematically better than the other.

But in the clinical domain, translation-based methods can allow to leverage domain-specific language models, whereas the multilingual models used for CTL are all from the general domain. This leads to our last open question:

Open Question 5.1. *How do translation-based methods and cross-lingual transfer compare in the clinical domain? Is it possible to leverage domain-specific language models with translation-based methods?*

Applying CTL to the clinical domain requires two kinds of adaptation for a given model: adaptation to a specific domain and adaptation to a specific language (CTL). While CTL is extensively studied in this thesis, domain adaptation is kept in its simplest form in our contribution: the experiments proposed in this thesis only rely on domain adaptation through pre-training, where monolingual LMs are pretrained on domain-specific data, like PubmedBERT (Gu et al., 2021), trained on English abstracts of biomedical scientific articles, and DrBERT (Labrak et al., 2023), trained on French clinical text data.

Several other domain adaptation techniques exist for monolingual word embeddings (Ramponi and Plank, 2020), and have been applied in the clinical domain (El Boukkouri, 2021), leveraging external knowledge that can be found either in the form of structured knowledge bases or in the form of large corpora of unlabeled texts. This thesis focuses on the latter, with models like PubmedBERT and DrBERT, because the "multilingual adaptation" of general-domain multilingual models like mBERT and XLM-R is also done trough pre-training. Therefore this thesis compares monolingual domain-specific pre-training with multilingual general-domain pre-training, to see which is more useful for a cross-lingual domain-specific task. Unfortunately, pre-trained LMs that are simultaneously mulitilingual and domain-specific do not exist yet. This thesis still finds evidence that they could be useful in a cross-lingual and domain-specific setting and advocates for more research that simultaneously tackles domain adaptation and cross-lingual transfer.

5.3 Conclusion

This chapter summarizes the gaps in the literature that this thesis intends to address. The following chapters detail the contribution of this thesis and the experiments that address some of those gaps.

Chapter 6

Language-agnostic Static Embeddings and the Impact of Code-switching

This chapter investigates the possibility of having language-agnostic static embeddings. It shows that a mainstream word embedding technique like FastText (Bojanowski et al., 2017) does not provide an effective multilingual alignment when trained, like mBERT, on a multilingual corpus without any explicit cross-lingual signal.

However, such a training method still provides a better-than-random alignment between language representations. This chapter analyses the cause of this alignment and shows the impact of code-switching, where monolingual corpora actually contain many occurrences of words from other languages, as exemplified in Figure 6.1.

Example 1 : 1999年歐洲歌唱大賽(eurovision song contest 1999) 為歐洲歌唱大賽之第44屆比賽
Example 2 : as a result , " li " (禮) , meaning " ritual " or " etiquette , "governed the conduct of the nobles , whilst " xing " (刑) , the rules of punishment
Example 3 : 是一款由鬼游(ghost town games) 公司, team 17 行的烹模游. 玩家通多人合作或多角控制, 控制多游角色挑各种房里的机

Figure 6.1 – Example of code-switching

Unfortunately, the reported experiments also indicate that those conclusions on static embeddings do not extend to contextualized ones, namely code-switching does not seem to be the most important factor in the success of multilingual contextualized embeddings.

For all experiments in this chapter, except mention of the contrary, two corpora are used: a dump of Wikipedia in English and one in Chinese. The rationale behind this choice is to use a general-domain corpus that is often used for building static embeddings, and even contextualized ones (like mBERT), in two high-resource languages that do not share the same script. Using different scripts allows us to identify code-switching situations more easily, by simply finding words written in one script surrounded by words written in the other script, although those code-switching situations would also include transliterations.

6.1 Implementation details

This list the resources used for implementing the experiments of this chapter.

6.1.1 Evaluating multilingual alignment

To evaluate the multilingual alignment of various language-agnostic static embeddings, this thesis relies on bilingual dictionaries provided by Lample et al. (2018). Dictionaries as well as evaluation scripts can be obtained in a github directory[12].

6.1.2 Evaluating cross-lingual transfer

To evaluate the impact of code-switching on cross-lingual transfer, this thesis relies on a single downstream task: the universal depency dataset for Part-of-Speech tagging (POS-tagging). The English training set used is the treebank English-EWT is used, while for evaluating of Chinese, the Chinese-GSD is used.

6.1.3 Extracting code-switching situations

For extracting English and Chinese words based on scripts, the following character unicode ranges are used:

- English: `[a-zA-Z]`
- Chinese: `[\U4e00-\U9fff\U3400-\U4dbf\U00020000-\U0002a6df\U0002a700-\U0002ebef\U00030000-\U000323af\Ufa0e\Ufa0f\Ufa11\Ufa13\Ufa14\Ufa1f\Ufa21\Ufa23\Ufa24\Ufa27\Ufa28\Ufa29\U3006\U3007][\Ufe00-\Ufe0f\U000e0100-\U000e01ef]?`

The regular expression in English is a simple range. For Chinese (Han script), a more complex regular expression is used as it must take into account variation selectors that can be added after an ideogram. The whole regular expression was obtained from `https://ayaka.shn.hk/hanregex`.

6.1.4 Learning static embeddings

To learn a static embedding from a given corpus, this thesis relies on FastText (Bojanowski et al., 2017), using the version that is the most recent at the time of writing: v0.9.2. The skipgram embeddings is used with default parameters (window size of 5, learning rate of 0.05, five epochs among other parameters) except for the dimension of the produced embeddings which are set to 300 as other ready-made embeddings used throughout this thesis have this same size.

6.1.5 Learning contextualized embeddings

To learn contextualized embeddings from a given corpus, this thesis relies on a distilmBERT model (Sanh et al., 2019) with weights that have been reset to random values. This allows us to use a ready-made tokenizer by keeping the vocabulary of the original model. It has 6 layers, 768 dimensions, and 12 attention heads and a vocabulary of 110k words and subwords.

The `transformers` library (version 4.24.0) is used to pre-train the model and to fine-tune it for cross-lingual transfer. Pre-training is done for 8 millions steps of batches of 4 text chunks of length 128. Other learning-related parameters of the training use the default parameters of the `TrainingArguments` class in the `transformers` library.

[12] `https://github.com/facebookresearch/MUSE`

embedding	normal BLI	easy BLI
unaligned embeddings	0.00%	0.00%
language-agnostic	0.00%	34.53%
supervised (Procrustes)	43.3%	74.7%

Table 6.1 – Results of Bilingual Lexicon Induction (BLI) for a language-agnostic bilingual word embeddings compared to a supervised baseline of aligned word embeddings. "normal BLI" is the result of the BLI when the search space is the whole target vocabulary, whereas "easy BLI" is the accuracy when the search space is reduced to the vocabulary of the bilingual dictionary.

6.2 Training a static multilingual embedding without cross-lingual signal

To learn a language-agnostic static embeddings, this section simply relies on a monolingual embedding technique, FastText (Bojanowski et al., 2017), as detailed in Section 6.1.4, and it is trained on a corpus built by concatenating a Wikipedia dump in English and one in Chinese. The Chinese one is repeated such that it constitutes near of half of the pretraining data.

As shown on Table 6.1, when evaluated on Bilingual Lexicon Induction (BLI) using MUSE bilingual dictionaries (cf. Section 6.1.1), it provides a top-1 accuracy of 0.0%, while a supervised alignment of monolingual embeddings can reach more than 40% for the English-Chinese pair. At first sight, language-agnostic training does not seem to provide any cross-lingual abilities to a static embedding, contrary to what the literature suggests for contextualized embeddings (cf. Chapter 4).

However, the obtained language-agnostic static embedding is actually still slightly aligned. If the search space for the nearest-neighbor search is reduced, BLI accuracy increases. For example, if we reduce the search space to the words present only in the 1,500 pairs of the bilingual dictionary, the BLI accuracy increases to 34,53%, while the unaligned baseline still produces a 0% accuracy. However, the language-agnostic embedding still lags behind the competitive supervised baseline.

But what in the training data can provoke this slight alignment? FastText with default parameters relies on a context window of 10 words (5 to the left and 5 to the right of the input word). Hence, there must be words from both languages that appear in the same context window. This thesis investigates two ways in which words from different language can share the same context:

- Common words: words that belong to the vocabulary of both languages. In our case, it will be only punctuations, digits, and other special characters. This presence of common words between languages, even the most distant ones has been hypothesized to be the reason for the success of multilingual contextualized models (Pires, Schlinger, and Garrette, 2019).
- Code-switching: as already shown in Figure 6.1, the monolingual corpora might not be totally monolingual and might include words from other languages (even in different scripts) for various reasons: explaining a concept from another culture, translating a foreign quote, describe a foreign geographical place, etc...

The following sections tries to quantify those two phenomenon, while the section after that investigate which of them, if any, is responsible for the slight alignment observed in a language-agnostic static embedding.

6.3 Quantifying naturally-occuring code-switching

To evaluate the amount of code-switching in a corpus, this section must rely on a dictionary or rather a list of words that are guaranteed to originate from a given language. Indeed if one relies only on different scripts, as will be done in later sections for other purposes, one might have an issue with the precision of the code-switching retrieval as the same script can be used by different languages. Using a dictionary can lack a bit of recall, as a dictionary can hardly contain all the

lang	tokens	token contamination			code-switching		
		coverage (%)	count	count digits	coverage (%)	count	count digits
ar	229M	44.9	1,043,396	6,511,347	38.0	486,764	6,360,450
ru	685M	55.1	5,237,773	26,063,394	50.7	4,158,232	25,637,900
zh	319M	47.6	1,720,247	3,220,332	39.4	1,174,912	3,117,309

Table 6.2 – Presence of words from an English dictionary in three non-English Wikipedia dumps. Contamination considers all words that were found in the corpus, and code-switching considers them only if they are in the vicinity of a word written in the non-English script. "coverage" is the proportion of the dictionary that was found and "count", the number of single occurrences. Occurrences of digit tokens are given for comparison.

ranks	ar	ru	zh
1-10	100.0	100.0	100.0
11-100	100.0	100.0	100.0
101-1,000	99.3	99.1	99.7
1,001-10,000	86.8	93.2	90.0
≥10,001	30.9	45.8	32.1

Table 6.3 – Proportion (in %) of English words in a dictionary covered by code-switching situations, split by buckets of frequency rank. e.g. line "1-10" indicates the proportion of the ten most frequent words of the dictionary that are covered by code-switching situations in each non-English language.

vocabulary used in English with all their inflections. But if the dictionary is comprehensive enough, it should provide a good lower bound of the number of code-switching situations.

This section uses the `3of6game` dictionary from the 6th version of the `12dicts`[13]. This dictionary contains 64,662 words. It was chosen because it is said to be oriented towards common words and was manually checked for errors, which should reduce the chance of the dictionary itself being polluted by code-switching. It is obtained from 6 advanced learners' ESL dictionaries, and contains American and British English, with inflections and neologisms.

This thesis differentiates token contamination and code-switching. Token contamination is simply the fact of finding an English token in a non-English corpus, but is not necessarily a code-switching situation, as the whole sentence might actually be written in English. With real code-switching, the English word must be found in the same context as a non-English one (identified with its script). A code-switching situation is thus also a token contamination situation. But the reciprocal is not necessarily true.

Table 6.2 shows that code-switching is present in all the tested datasets. From around 500,000 situations in Arabic to more than 4 millions in Russian. This is a small fraction of the hundreds of millions of tokens present in each corpus. But, to comprehend what the frequency of code-switching represents, Table 6.2 shows that code-switching is 3 to 15 times rarer than digits. This goes on to show that code-switching is not an exceptional occurrence in a monolingual corpus like Wikipedia.

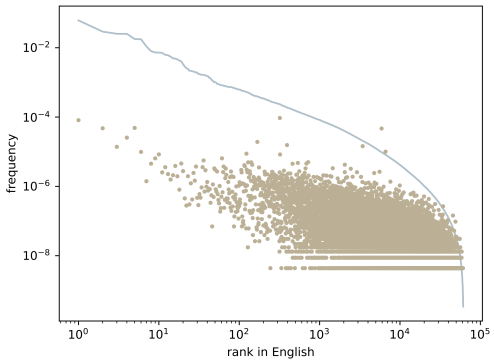
While code-switching is relatively scarce, it however covers an important portion of the English vocabulary. Indeed, Table 6.2 shows that code-switching situations cover up to half of the English dictionary. A breakdown by frequency shows that the most frequent words are almost all involved in code-switching situations, as shown in Table 6.3.

Figure 6.2 compares the frequency of English words in the English corpus with their frequency in a non-English corpus. It shows that the frequency of a code-switched word rarely exceeds 10^{-4}, with frequent words in English being generally more frequently code-switched than infrequent ones.

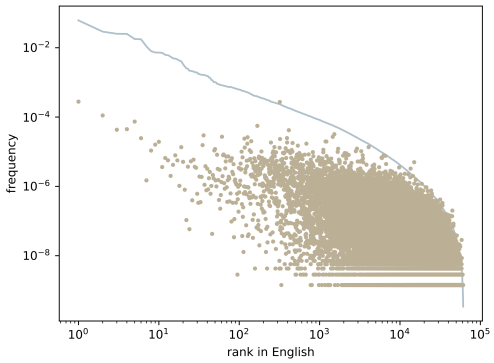
The results of this section suggest that code-switching, despite being infrequent, amounts to a non-negligible number of code-switched tokens in a large corpus which covers a large part of the most frequent words from the code-switched language.

Code-switching seems significantly less frequent than the use of common subwords like punctua-

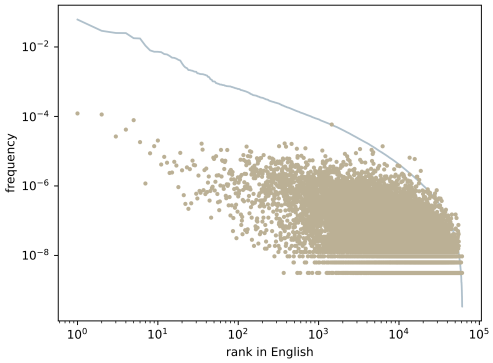
[13] http://wordlist.aspell.net/12dicts-readme



(a) Arabic



(b) Russian



(c) Chinese

Figure 6.2 – Frequency of words from an English dictionary in the English corpus (line) and non-English one (dots) according to the rank in frequency in the English corpus.

tion marks or digits. However, it seems to cover a larger range of the vocabulary. The next section will confirm what might already be suspected: code-switching is what induces a slight alignment in the language-agnostic embedding from the previous section.

6.4 The impact of code-switching on language-agnostic static embeddings

To measure the impact of code-switching on language-agnostic static embeddings, this section performs an ablation analysis on the embedding proposed earlier.

As shown in Table 6.4, the ablation is performed on the two monolingual corpora used for training the embeddings. In the `original` version, nothing particular is done, the monolingual corpora are kept as is. In `no-code-switch`, code-switching is removed based on scripts. In practice, tokens of Latin letters are removed from the Chinese corpus and Chinese ideograms are removed from the English one. On the other hand `no-special` removes all potentially common tokens. For simplicity, it removes all characters that are neither a Latin letter nor a Chinese ideogram from the two corpora. Finally `script-only` removes code-switching as well as common words. For that, only ideograms are kept in the Chinese dataset while only Latin letters are kept in the English one.

Table 6.5 shows the results of the ablation analysis. It demonstrates that when code-switching

`original`	1999年歐洲歌唱大賽(eurovision song contest 1999)為歐洲歌唱大賽之第44屆比賽
`no-code-switch`	1999年歐洲歌唱大賽(1999) 為歐洲歌唱大賽之第44屆比賽
`no-special`	年歐洲歌唱大賽eurovision song contest 為歐洲歌唱大賽之第屆比賽
`script-only`	年歐洲歌唱大賽為歐洲歌唱大賽之第屆比賽

Table 6.4 – The four variants of the corpus for the ablation analysis.

training corpus	easy BLI
`original`	34.53%
`no-code-switch`	2.44%
`no-special`	38.07%
`script-only`	0.00%

Table 6.5 – Results of the ablation analysis with "easy BLI" (reduced search space).

is removed (`no-code-switch`), the alignment is reduced. On the other hand, removing the punctuation and digits does not reduce alignment, it even increases it (`no-special`). Without both (`script-only`), when the vocabularies of both languages truly have no intersection, the accuracy unsurprisingly drops to zero. Nevertheless code-switching seems to be more important than sharing punctuation characters or digits for learning good multilingual representations in a language-agnostic manner.

6.5 Could code-switching explain contextualized embedding abilities?

Previous sections have demonstrated that code-switching is a naturally-occurring cross-lingual signal that can be picked up by language-agnostic static embedding. However, it is not because code-switching explains the slight multilingual alignment of some static embeddings that it also explains the cross-lingual transfer abilities of larger multilingual models. This section investigates whether code-switching is an important signal for a deeper language-agnostic model.

A similar ablation analysis as the previous section is performed on a small Transformer encoder. The model and its pre-training are already described in Section 6.1.5.

The pre-training is performed either on the `original` or on the `no-code-switch` version of the bilingual dataset. It is trained for eight million steps. Then the model is fine-tuned in English on POS-tagging and evaluated on the same task in Chinese, as well as on an English validation set.

Results averaged over five seeds are in Table 6.6. They show that with and without code-switching (`original` and `no-code-switch`), pre-training leads to better results than a randomly-initialized model without pre-training. Removing code-switching from the pre-training data decreases the downstream cross-lingual abilities significantly, without harming the same-language accuracy (en). Nevertheless, code-switching alone might not be sufficient to explain the success of language-agnostic

pre-training	en	zh
`original`	89.15%±0.45	28.34%±1.80
`no-code-switch`	89.12%±0.10	26.08%±1.83
without pre-training	79.85%±0.45	18.51%±1.56

Table 6.6 – Ablation analysis on a small Transformer encoder. The model is evaluated on POS-tagging, after fine-tuning in English.

Transformers.

6.6 Conclusion

Training a static embedding without any explicit cross-lingual signal does not provide a good multilingual alignment as it would for a multilingual Transformer. However, such language-agnostic training still provides a better-than-random multilingual alignment, showing that Transformers might not be the only model architecture for building multilingual models without any explicit cross-lingual signal.

In language-agnostic static embeddings, the slight emergence of multilingual abilities can be entirely explained by the presence of code-switching as shown by an ablation experiment. However, code-switching alone does not seem to explain the abilities of larger cross-lingual models, although it might play a part. Nevertheless, code-switching is still a valuable cross-lingual signal. Further experiments performed in the context of this thesis show that it can be used to align multilingual embeddings (cf. Appendix 3).

Chapter 7

Investigating the limitations of multilingual static embeddings

The literature suggests that multilingual Transformers provide better cross-lingual transfer results than models based on aligned static embeddings. Moreover, the next chapters of this thesis show that alignment and cross-lingual transfer are strongly linked and that the alignment in language-agnostic contextualized representations is competitive with explicitly aligned static embeddings. At the same time, using language-agnostic training for static embeddings, similar to multilingual Transformers, does not provide well-aligned embeddings. Despite the superiority of Transformers in a multilingual setting, the current chapter investigates multilingual static embeddings.

This thesis argues that multilingual static embeddings are still a worthy object of study. First of all, they require far less compute to train than Transformers. Inference is also orders of magnitude simpler since computing the static embedding of a given word consists in querying a look-up table. Additionally, for tasks like Named Entity Linking (NEL), where the context of a word is not necessarily provided, using a static embedding might make more sense than using a contextualized one, by definition.

Nevertheless, the literature shows that multilingual static embeddings namely those obtained in an unsupervised manner, have strong limitations (Sogaard et al., 2019). This chapter provides a better understanding of those limitations, focusing on two of them.

The first limitation is that they fail to provide alignment for some pairs of distant languages. This chapter shows that the initialization proposed by mapping-based methods plays a great role in this failure. It also demonstrates that code-switching might help alleviate this issue.

The second studied limitation is that embeddings from different domains cannot be aligned with most existing unsupervised methods. The later sections of this chapter show that contrary to distant languages, the failure of alignment can be linked to some inadequacy of the usual self-learning procedure. Namely, it suggests that learning an alignment between subsets of the vocabularies instead of the entire ones might be a good lead for solving the issue.

In this chapter, the multilingual embeddings are only evaluated through BLI. While it was shown that selecting embeddings based on BLI can hurt downstream performances (Glavaš et al., 2019), this thesis focuses on the cases where alignment fails. Therefore, a low BLI score should suffice to identify those cases.

7.1 The failure of initialization with some languages

Søgaard, Ruder, and Vulić (2018) show that a typical mapping-based alignment method fails for distant languages. More specifically, they show that the adversarial mapping learned with MUSE (Lample et al., 2018) works well for aligning embeddings from the same domain in English and Spanish, but not for the English-Finnish and English-Hungarian language pairs. As a follow-up, this section extends the analysis to different methods and more language pairs. It shows that some

method	initialization	self-learning	
		dictionary inference	mapping inference
Existing unsupervised methods			
MUSE	adversarial	nearest-neighbor	Procrustes
WP	graph-matching	Wasserstein	Procrustes
Vecmap	graph-matching-inspired heuristics	nearest-neighbor with randomly dropping out pairs	Procrustes
Methods with VecMap self-learning for controlled comparison			
Vecmap identical	dictionary of identical words	Vecmap self-learning	
Vecmap w/ MUSE	MUSE initialization	Vecmap self-learning	
Vecmap w/ WP	WP initialization	Vecmap self-learning	
Supervised baselines			
Supervised w/ self-learning	training dictionary	Vecmap self-learning	
Supervised	training dictionary	*no self-learning*	

Table 7.1 – List of baselines used in this chapter

pairs of languages are harder to align, even with a supervised method. However, comparing different unsupervised methods to a supervised baseline reveals that even when two embeddings can be aligned with a supervised method, unsupervised approaches sometimes fail to find any alignment. The reported experiments also demonstrate that those failure cases can largely be explained by a failure of the initialization of those methods, but in the case of embeddings from different domains, the following sections will also show that there is an additional issue due to the discrepancies in domain vocabulary.

7.1.1 Compared methods

This chapter focuses on mapping-based approaches that learn an orthogonal mapping. It compares several unsupervised methods to supervised baselines, with and without self-learning, that also rely on an orthogonal mapping.

As already detailed in a previous chapter (Section 3.2.3), unsupervised mapping-based methods work in two steps:

- the initialization: which aims at learning a seed dictionary or an initial mapping;
- the self-learning procedure: which iteratively improves the alignment by learning a new dictionary from a previous mapping and then a new mapping from a previous dictionary.

We compare three existing unsupervised methods: Wasserstein-Procrustes (Grave, Joulin, and Berthet, 2018), MUSE (Lample et al., 2018), and Vecmap (Artetxe, Labaka, and Agirre, 2018a).

Wasserstein-Procrustes (**WP**) (Grave, Joulin, and Berthet, 2018) is a method relying on optimal transport. The initial dictionary is provided through the convex relaxation of a graph-matching problem between the graphs, for each monolingual embedding, of similarities between each word. Self-learning is then performed. At each step, a new mapping is learned from a given dictionary with Procrustes as in most other methods. A new dictionary is obtained from a given mapping by solving an optimal transport problem using Wasserstein distance.

MUSE (Lample et al., 2018) relies on adversarial learning. A linear mapping is trained to maximize the loss of a discriminator that is simultaneously trained to distinguish embeddings from both languages that are being aligned. The mapping W is orthogonalized at each step using the following update:

$$W \leftarrow (1+\beta)W - \beta(WW^T)W \tag{7.1}$$

β is a hyper-parameter, fixed to 0.01 following (Lample et al., 2018). The obtained mapping is then refined with self-learning. Each new mapping is obtained with Procrustes. Each new dictionary is obtained through a nearest-neighbor search.

VecMap (Artetxe, Labaka, and Agirre, 2018a) relies, like WP, on graph-matching for initialization: each word is represented by a vector containing the distance to all other words. After taking the square root of each embedding matrix, sorting the values in each vector, and normalizing them, a nearest-neighbor search provides the initial dictionary. Self-learning then consists of Procrustes for learning each new mapping and nearest-neighbor search for learning each new dictionary.

Søgaard, Ruder, and Vulić (2018) showed that fully unsupervised mapping-based methods can fail in certain conditions, namely when languages are distant. They obtain better results using a seed dictionary built with identical words found in both vocabularies instead of one resulting from graph-matching algorithms or adversarial mapping that might rely too heavily on the need for isometry between embeddings. This initialization is used with VecMap self-learning to compare with ours and VecMap.

To perform a controlled comparison of different initialization methods, two other methods are designed by simply replacing the initialization in Vecmap by the initialization of either MUSE or WP.

As baselines, two supervised methods are used. They are trained on a bilingual dictionary of the same origin as the evaluation one, but with 1,500 pairs that are distinct from the evaluation ones. The first supervised method simply uses Procrustes to learn a mapping from the given dictionary, while the second one leverages self-learning. While the former serves as an upper baseline to compare with the unsupervised methods, the latter allows us to evaluate the effect of self-learning by comparison with the other supervised baseline.

All the baselines are listed in Table 7.1.

7.1.2 Results of mapping-based methods for different language pairs

The methods described in the previous section are evaluated in Table 7.2. This section details what can be learned from those results.

Some languages are easier to align than others. Results of the supervised method (without self-learning) show an inherent limitation of alignment based on orthogonal mapping which echoes the literature on the lack of isometry between embeddings of distant languages (Søgaard, Ruder, and Vulić, 2018). Since Procrustes provides a closed-form solution for the supervised problem, it means that it is unlikely that a better alignment can be found in the set of possible solutions provided by orthogonal matrices. However, it must be noted that it does not necessarily mean that a better alignment doesn't exist at all since the supervised method is evaluated on a distinct evaluation dictionary. Indeed, the supervised method with self-learning and even some unsupervised baselines like Vecmap can outperform the Procrustes baselines by a few points, as in French.

Distant pairs of languages are hard to align, but it seems that it is not the only factor of failure. Namely, the worst performing pair is English-Afrikaans, where Afrikaans is a language derived from Dutch and is thus not the most dissimilar language with respect to English as both are West Germanic languages. However, Afrikaans is a low-resource language which might be another factor for the failure of alignment methods.

The self-learning from Vecmap is robust. The two supervised baselines, with and without self-learning, reach comparable results. This suggests that the self-learning procedure proposed with Vecmap does not harm the alignment obtained with the seed dictionary. This might not seem surprising, but later in this chapter, it will be shown that this self-learning can be detrimental in the cross-domain setting. But here, this self-learning approach is efficient and provides better results than those of MUSE and WP, since Vecmap with the initialization from either MUSE or WP provides better results than MUSE or WP themselves with their original self-learning procedure. This is consistent with the findings of Hartmann, Kementchedjhieva, and Søgaard (2019)

method	fr	it	hu	ja	ru	vi	fi
MUSE	$82.2_{\pm 0.2}$	$77.5_{\pm 0.4}$	$49.9_{\pm 1.8}$	$0.0_{\pm 0.0}$	$41.7_{\pm 2.9}$	$0.0_{\pm 0.0}$	$41.9_{\pm 2.4}$
WP	$81.0_{\pm 0.2}$	$74.3_{\pm 1.0}$	$43.5_{\pm 1.7}$	$3.8_{\pm 6.8}$	$36.9_{\pm 1.4}$	$0.0_{\pm 0.0}$	$0.2_{\pm 0.2}$
VecMap	$\mathbf{82.4}_{\pm 0.1}$	$\underline{79.0}_{\pm 0.0}$	$56.1_{\pm 0.2}$	$11.2_{\pm 19.1}$	$\mathbf{49.1}_{\pm 0.4}$	$0.7_{\pm 0.4}$	$\mathbf{50.3}_{\pm 0.2}$
Vecmap w/ MUSE	$\mathbf{82.4}_{\pm 0.0}$	$\mathbf{79.1}_{\pm 0.1}$	$56.5_{\pm 0.4}$	$0.0_{\pm 0.0}$	$47.5_{\pm 2.5}$	$9.6_{\pm 19.1}$	$49.5_{\pm 0.6}$
Vecmap w/ WP	$\underline{82.3}_{\pm 0.0}$	$\underline{79.0}_{\pm 0.1}$	$56.7_{\pm 0.2}$	$20.3_{\pm 23.1}$	$44.8_{\pm 3.3}$	$0.2_{\pm 0.1}$	$0.0_{\pm 0.1}$
Vecmap identical	$\underline{82.3}_{\pm 0.1}$	$\underline{79.0}_{\pm 0.1}$	$\mathbf{57.0}_{\pm 0.2}$	$\mathbf{48.4}_{\pm 0.7}$	$\underline{48.9}_{\pm 0.2}$	$\mathbf{48.6}_{\pm 0.6}$	$\underline{50.2}_{\pm 0.2}$
supervised w/ self-learning	$82.4_{\pm 0.0}$	$79.0_{\pm 0.1}$	$57.1_{\pm 0.2}$	$48.7_{\pm 0.2}$	$49.1_{\pm 0.2}$	$47.9_{\pm 0.4}$	$50.3_{\pm 0.2}$
supervised	81.6	78.8	57.2	52.8	52.7	49.8	49.2

method	zh	ar	af	bs	th	tl	ta
MUSE	$0.0_{\pm 3.3}$	$30.9_{\pm 3.3}$	$4.5_{\pm 8.9}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	-*	$0.0_{\pm 0.0}$
WP	$0.6_{\pm 0.8}$	$10.7_{\pm 9.9}$	$23.5_{\pm 1.1}$	$0.5_{\pm 0.6}$	$0.0_{\pm 0.0}$	$3.0_{\pm 3.0}$	$0.0_{\pm 0.0}$
VecMap	$0.0_{\pm 0.0}$	$36.1_{\pm 1.8}$	$33.7_{\pm 1.0}$	$0.1_{\pm 0.2}$	$0.0_{\pm 0.0}$	$0.1_{\pm 0.1}$	$0.1_{\pm 0.1}$
Vecmap w/ MUSE	$6.5_{\pm 12.6}$	$\underline{39.5}_{\pm 0.7}$	$7.2_{\pm 14.0}$	$0.1_{\pm 0.2}$	$0.0_{\pm 0.0}$	-*	$3.0_{\pm 5.9}$
Vecmap w/ WP	$0.0_{\pm 0.1}$	$31.9_{\pm 15.9}$	$\underline{34.7}_{\pm 0.4}$	$0.2_{\pm 0.1}$	$0.0_{\pm 0.0}$	$8.7_{\pm 6.8}$	$0.0_{\pm 0.0}$
Vecmap identical	$\mathbf{36.8}_{\pm 0.8}$	$\mathbf{39.8}_{\pm 0.3}$	$\mathbf{34.9}_{\pm 0.4}$	$\mathbf{28.3}_{\pm 0.8}$	$\mathbf{24.0}_{\pm 0.5}$	$\mathbf{19.4}_{\pm 0.4}$	$\mathbf{17.8}_{\pm 0.3}$
supervised w/ self-learning	$38.1_{\pm 0.4}$	$39.5_{\pm 0.1}$	$34.5_{\pm 0.4}$	$28.5_{\pm 0.3}$	$24.5_{\pm 0.2}$	$19.1_{\pm 0.6}$	$18.3_{\pm 0.5}$
supervised	43.3	43.0	34.8	28.9	25.1	22.8	20.2

*MUSE was not applicable for this language with default parameters because it requires a vocabulary of at least 75,000 words and the Wikipedia embedding has less.

Table 7.2 – Results of many alignment methods for several language pairs. The evaluation method is Bilingual Lexicon Induction (BLI) with dictionaries from Lample et al. (2018). The reported accuracy is the top-1 accuracy for the CSLS criterion (Joulin et al., 2018) for translation from English to the target language. Languages are sorted by decreasing results for the supervised method. With the exception of "supervised" which is deterministic, all methods were applied 5 times with different seeds and the standard deviation is reported alongside the mean. Bold indicates the best unsupervised baseline for a given language and values that are within its standard deviation are underlined.

method	results for different seeds				
WP	14.9	5.7	28.0	5.1	0.0
MUSE	34.1	33.9	26.5	32.4	27.3
Vecmap	37.8	37.4	35.9	33.2	37.9
Vecmap w/ MUSE	39.5	39.5	39.9	40.2	38.3
Vecmap w/ WP	0.1	39.6	40.5	39.3	39.9
Vecmap identical	40.3	39.7	39.9	39.4	39.8

Table 7.3 – Breakdown of the BLI accuracy for each of the tested random seeds for the English-Arabic language pair. Each column represents a different random seed used for the algorithms.

showing that the success of the most recent methods like VecMap is mainly due to their proposed self-learning.

This comparison allows us to answer the following open question:

Open Question 3.2. *Could VecMap self-learning algorithm be used and provide more stable results with other kinds of initialization like adversarial methods?*

Vecmap self-learning seems to provide a better stability of the method. Table 7.3 indeed shows that WP and MUSE can provide very different results according to the random seed used, while using the same initialization with Vecmap self-learning provide a better and more stable alignment. In the end, it seems that it's the proposed self-learning procedure that makes the success and the robustness of Vecmap rather than its initialization.

Not all unsupervised methods fail on the same language pairs. It's not because two embeddings can be aligned that existing unsupervised methods will find an effective alignment. Hartmann, Kementchedjhieva, and Søgaard (2018) have already shown that the typical optimization landscape for learning a bilingual mapping is filled with local optima. For example, Japanese (ja) and Russian (ru) can be aligned to English with a supervised method with a similar accuracy (respectively 52.8% and 52.7%). However, most unsupervised methods largely fail on Japanese while they provide fair results for Russian, that are comparable with the supervised baselines. And when one unsupervised method succeeds, another might fail. For example, the WP initialization is the only one that dramatically fails for the English-Finnish (fi) alignment, while MUSE is the sole one that fails for Afrikaans (af). On pairs that are more difficult to align, existing unsupervised initialization methods are unreliable.

7.1.3 Conclusion on distant and low-resource language pairs

This section has identified three factors to mitigate the failure of unsupervised mapping-based methods:

- The **quality of the initialization** is probably the most important one since some initialization methods make the alignment systematically fail on specific languages, whatever the self-learning procedure may be.
- The **robustness of the self-learning procedure** also allows more stable results across seeds. This thesis shows that the self-learning procedure from Vecmap might be the more robust in that sense.
- Finally, the **isometry of the initial embeddings** can hamper the results. It is not the main reason for the near-zero accuracy of unsupervised methods, since a supervised method can obtain fair results on all pairs of languages. However, this supervised mapping-based method materializes a glass ceiling that isometry-based unsupervised methods might not be able to significantly overcome.

The results of "Vecmap identical", proposed by Søgaard, Ruder, and Vulić (2018), demonstrate that the first issue can be solved, as its initialization allows to match the supervised baseline with self-learning. In this setting, the issue with the robustness of the self-learning procedure seems to have been solved by Vecmap (Artetxe, Labaka, and Agirre, 2018a). Finally, the lack of isometry between the monolingual embeddings cannot be overcome with the mapping-based methods used, since they learn orthogonal mappings. One must turn to other methods, supervised or unsupervised, if one wants to overcome this issue.

7.2 The failure of self-learning for different domains

While the previous section provides reasons for the failure of unsupervised mapping-based methods for distant pairs of languages, the following one takes interest in another failure case: when

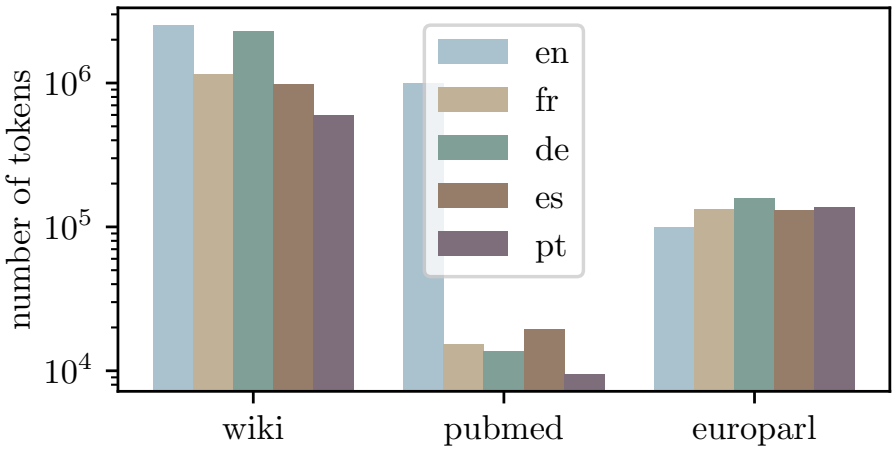


Figure 7.1 – Number of distinct tokens in each corpus by language.

embeddings come from different domains. Søgaard, Ruder, and Vulić (2018) have shown that aligning monolingual embeddings in distinct languages and from different domains made MUSE largely fail. The following will show that the reasons for failure in a cross-domain setting are not the same as for distant language pairs.

7.2.1 A biomedical cross-domain multilingual setting

This section is motivated by a need for effective cross-domain multilingual representations. Indeed, Domain-specific data, such as scientific publications, can be rare in languages other than English. Figure 7.1 shows that while a general domain corpus like Wikipedia can contain a fair amount of data for several languages, the imbalance between languages is dramatically higher for a corpus of a more specific domain like Pubmed, which contains millions of abstracts of scientific publications from the biomedical domain. Thus, to build a multilingual biomedical embedding, one could rely on Pubmed for English and fall back to more general-domain data in other languages.

This section uses the same experimental setting as for distant languages, but instead of aligning an English embedding from Wikipedia with Wikipedia embeddings in other languages, this section replaces the English embedding by one trained on Pubmed. It is obtained by running FastText (Bojanowski et al., 2017) on a Pubmed dump with the same default parameters used for obtaining the Wikipedia embeddings.

7.2.2 Comparing existing methods with different domains

Table 7.4 shows the BLI accuracy of different baselines for a cross-domain alignment between Pubmed embedding in English and a target language embedding from Wikipedia. This section analyzes those results.

Unsupervised baselines largely fail in a cross-domain setting. With the exception of the weakly-supervised baseline "Vecmap identical" which uses pairs of identical words as a seed dictionary, all unsupervised baselines obtain zero or near-zero accuracy for all tested language pairs. However, since a supervised baseline can allow some level of alignment, it suggests that a better unsupervised alignment is possible. The question then arises: which part of the unsupervised method is failing in the cross-domain setting: the initialization or the self-learning procedure?

Self-learning is harming the results. This is suggested by the large gap between the simple supervised baseline and the one with self-learning. Both use the same training dictionary, but adding

method	fr	it	ru	hu
MUSE	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$
WP	$0.0_{\pm 0.1}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$
VecMap	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$
Vecmap w/ MUSE	$0.0_{\pm 0.1}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$
Vecmap w/ WP	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$
Vecmap identical	$20.1_{\pm 0.4}$	$12.0_{\pm 0.0}$	$1.5_{\pm 0.1}$	$1.1_{\pm 0.1}$
supervised w/ self-learning	$20.2_{\pm 0.6}$	$16.7_{\pm 0.2}$	$3.6_{\pm 1.0}$	$4.5_{\pm 0.4}$
supervised	42.7	36.1	25.1	23.2

method	ar	zh	fi	af
MUSE	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$
WP	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$
VecMap	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$
Vecmap w/ MUSE	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$
Vecmap w/ WP	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$
Vecmap identical	$6.8_{\pm 0.4}$	$0.0_{\pm 0.0}$	$0.1_{\pm 0.0}$	$0.1_{\pm 0.1}$
supervised w/ self-learning	$9.3_{\pm 0.7}$	$0.0_{\pm 0.0}$	$1.8_{\pm 0.5}$	$0.9_{\pm 0.2}$
supervised	22.1	20.6	18.5	14.9

Table 7.4 – Results of the chosen baselines in the cross-domain setting.

the self-learning makes the alignment worse, losing around 20 points of accuracy in all languages.

Initialization is still important. The gap between "Vecmap identical" and other unsupervised baselines suggest that, on top of the issues with self-learning, most initialization techniques fail to put the self-learning procedure in the right conditions to at least match the supervised method with self-learning.

Cross-domain embeddings are inherently hard to align. The results of the supervised baseline in this cross-domain setting are approximately half of those in the same-domain setting for the same languages. This suggests that cross-domain embeddings are hard to align, in the sense that they probably lack isometry.

To conclude this section, cross-domain alignment seems to suffer from the same issues as hard-to-align language pairs. However, the cross-domain setting adds another problem: self-learning is not working as it should. The following will provide a hypothesis as to why this problem arises in the cross-domain setting and provide some evidence to support it.

7.2.3 Partial isometry across domains

This thesis notices that the self-learning procedure of mapping-based methods usually tries to align the whole embedding together, or more precisely a large set of the most frequent words of each embedding, typically the 20,000 most frequent words in Vecmap. Yet, between two different domains, the distribution of the vocabulary may vary. Some words might be more frequent in one domain and less in the other, or even absent.

Different languages might lack isometry due to a lack of isometry at the local level. As pointed out by (Søgaard, Ruder, and Vulić, 2018), "the" in English can translate to "der", "die", or "das" in German depending on the context. This makes the isometry fragile on the local level. This thesis hypothesizes that a cross-domain setting creates an additional kind of lack of isometry: a global one. Typically, domain-specific vocabulary contains words that must not be aligned to any word from the other domain. Trying to align all words from the embedding of different domains might fail due to those words. A relatively good alignment might still be possible by discarding those domain-specific words, as exemplified in the toy example of Figure 7.2.

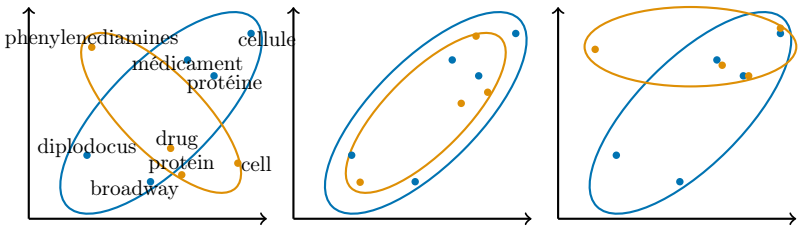


Figure 7.2 – Toy example showing different alignment of a domain (**orange**) with another (**blue**), the initial unaligned embeddings (left) do not align well when considering all words (center), but it might be possible to align them partially (right).

To support the proposed hypothesis, two experiments are performed: (1) the deviation from isometry is measured between different relevant subsets of the domain vocabularies, and (2) a visualization of such partial alignment is provided using a supervised method.

7.2.3.1 Experimentals details

For all experiments, this section leverages embeddings built on three different corpora (Wikipedia, PubMed, EuroParl) with five different languages (English, French, German, Spanish and Portuguese). PubMed is a collection of approximately 21 million biomedical abstracts[14] mainly written in English. EuroParl (Koehn, 2005) is a parallel corpus built from proceedings of the European Parliament. All embeddings were built with FastText (Bojanowski et al., 2017) with 300 dimensions, consistently with the previous chapter (cf. Section 6.1.4). Those from Wikipedia were obtained directly from the FastText website[15] and we trained FastText ourselves on PubMed and EuroParl using the official implementation. Figure 7.3 shows the vocabulary size for each embedding. Wikipedia embeddings as well as the English PubMed embeddings have a vocabulary size of the same order of magnitude. The other corpora are not comparable. This explains why cross-domain alignment might be needed as domain-specific data is sometimes lacking in languages other than English.

For measuring the deviation from isometry between various pairs of embeddings, this thesis relies on the GUDHI Python package (The GUDHI Project, 2020).

7.2.3.2 Measuring the deviation from isometry for various vocabulary subsets

To measure the deviation from isometry, this thesis relies on the approximation of the Gromov-Hausdorff (GH) distance, as Patra et al. (2019) showed that it correlates better than another metric with the performance of alignment methods (cf. Section 3.2.5).

For those experiments, embeddings from five different languages (English, French, German, Spanish, and Portuguese) are computed for two different domains: Wikipedia and Europarl. Additionally, the English Pubmed embedding is used, but no non-English Pubmed embeddings are used due to the lack of available data to train them.

The approximate GH distance is measured between various subsets of those embeddings. Those subsets are limited to 5,000 words for computability reasons and also because Patra et al. (2019) have used the same constant when showing the correlation of the metric with the ability to align word embeddings.

The approximate GH distance is computed between three kinds of subsets:

- The 5,000 most frequent words (in the original pre-training data) for both involved embeddings, which represents what the self-learning procedure is typically trying to align;

[14] Courtesy of the U.S. National Library of Medicine https://www.nlm.nih.gov/databases/download/pubmed_medline.html

[15] https://fasttext.cc/docs/en/pretrained-vectors.html

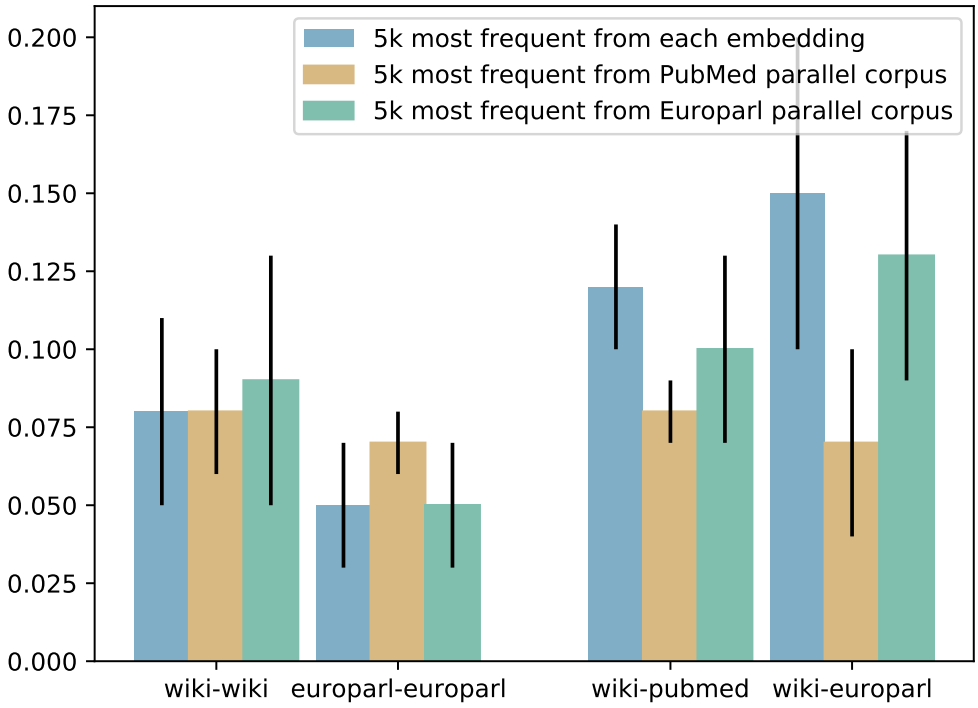


Figure 7.3 – Approximate GH distance measured between three subsets of vocabulary for four pairs of embeddings. The three different subsets are the most frequent words from each embedding (which self-learning is typically trying to align), and the most frequent words from parallel corpora: Pubmed and Europarl. Same-domain pairs of embeddings are compared on the left and cross-domain on the right. Segments show the standard deviation. Smaller values indicate better approximate isometry.

- the 5,000 most frequent words in each language in a parallel Pubmed corpus, which is supposed to contain comparable vocabulary since they are extracted from pairs of translated sentences;
- the 5,000 most frequent words in each language of a parallel Europarl corpus, which is similar to the previous option but with another domain.

The approximate GH distance is measured between different pairs of embeddings. Those pairs all involve English and another language, as only parallel corpora between English and other languages are available. Those pairs involve either embeddings from the same domain (Wikipedia with Wikipedia or Europarl with Europarl) or from different domains (Wikipedia with Pubmed or Wikipedia with Europarl). Results shown in Figure 7.3 shows the results averaged over all four non-English languages (French, German, Spanish and Portuguese).

When looking at the most frequent words of the embeddings themselves (**blue bars** in Figure 7.3), same-domain pairs (left) have smaller deviation from isometry than cross-domain ones (right). This is coherent with the fact that cross-domain embeddings are harder to align.

But when looking at the most frequent words in a parallel corpus drawn from Pubmed (**orange bars**), all pairs of embeddings have a small approximate GH distance, regardless of if they are cross-domain or same-domain. This suggests, as hypothesized, that the embeddings of comparable vocabularies have a lower deviation from isometry than the most frequent words from different domains.

However, this does not work perfectly with all supposedly comparable vocabulary. What worked for the Pubmed parallel corpus, works only to a lesser extent for the Europarl one (**green bars**). The deviation from isometry for cross-domain pairs is still reduced compared to the first subset, but this decrease is within the standard deviation and is not comparable with the one obtained with the Pubmed parallel corpus. But the Europarl embeddings and vocabularies might suffer from another factor that might increase the deviation from isometry: they are built from the transcription of spoken language whereas the others are based on written language. Between written and spoken language, the meaning and usage of words change, which might have harmed the results.

7.2.3.3 Observing the partial alignment with supervised methods

The previous experiment showed that some subsets of the vocabularies of cross-domain embeddings are easier to align than the subsets that are typically aligned using existing unsupervised methods. This suggests that there exist a partial approximate isometry between embeddings from different domains, but it doesn't show that this partial isometry can be effectively leveraged by existing alignment methods. To this end, the current section provides a visualization of the alignments provided by Vecmap and the supervised baseline in a same-domain and a cross-domain setting. This will allow us to visualize the partial alignment, learned by the supervised method in a cross-domain setting.

t-SNE (Maaten and Hinton, 2008), a dimensionality-reduction technique, is applied to four different English-French bilingual embeddings obtained either with Vecmap or the supervised baseline, and built either in a cross-domain or same-domain setting. This results in the Figure 7.4. The two colors represent the two languages involved, and black segments link 50 bilingual pairs of words randomly sampled from the training dictionary.

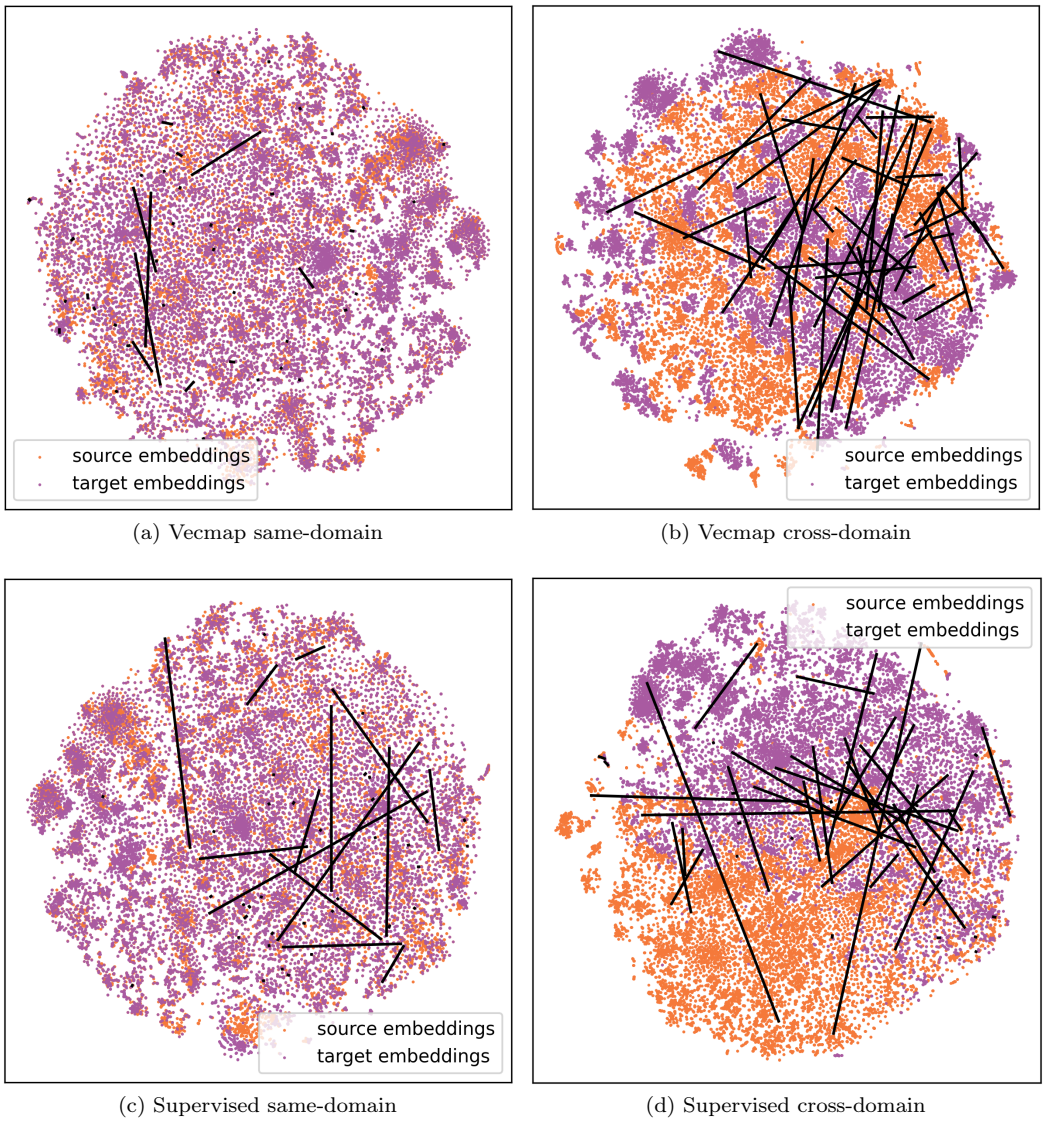
In the same-domain setting, Vecmap (7.4a) as well as the supervised baseline (7.4c) provide a good alignment. A small fraction of the 50 bilingual pairs of words are far apart and the entire monolingual embeddings seem to be well-aligned, with not only the global shapes of both monolingual embeddings being in the same area, but also small clusters overlapping across languages.

In the cross-domain setting, Vecmap (7.4b) has a higher fraction of visible black segments than the supervised baseline (7.4d), showing what the BLI accuracy already showed: translated pairs of words are better aligned with the supervised method while the unsupervised ones largely fail in a cross-domain setting. Moreover, Vecmap seems to provide a global alignment of both embeddings, but this global alignment does not work well locally. In many instances, both embeddings seem to almost "avoid" each other, explaining the BLI accuracy of 0. On the other hand, the supervised method does not provide the global alignment that the other three settings have. Instead, it seems to conform to the partial alignment hypothesized earlier, as schematized in the earlier toy example (Figure 7.2) as most words from the bilingual dictionary seem to be found in the overlap between the two aligned monolingual embeddings.

However, this lack of global isometry is not the only thing that hampers the supervised cross-domain alignment. Indeed the BLI accuracy is not on par with same-domain alignment and Figure 7.4d does not show the overlap of several small clusters as in the same-domain setting. It can be inferred that the issues that were observed with distant languages are also present, namely that there is a lack of local isometry between both embeddings, on top of this lack of global isometry that necessitates a partial alignment.

7.3 Conclusion

This chapter studies the failure of unsupervised mapping-based methods for multilingual embeddings in two cases: when aligning distant pairs of languages, and when doing so in a cross-domain setting. The comparison of various methods on language pairs that are hard to align shows that the quality of the initialization is crucial. Due to its instability, which was already shown in the literature, adversarial methods like MUSE can fail systematically on specific pairs of languages. Eventually, the only initialization that is robust enough across pairs of languages is to use words present in



(a) Vecmap same-domain

(b) Vecmap cross-domain

(c) Supervised same-domain

(d) Supervised cross-domain

Figure 7.4 – t-SNE for alignment alignment produced by Vecmap and the supervised baseline between French and English embeddings obtained either from the same domain (Wikipedia) or from different ones (Wikipedia for English and Pubmed for French). t-SNE is applied to the 20,000 most frequent words of both embeddings together. 50 bilingual pairs of words are sampled from the training dictionary and a black segment is drawn between their representations. For the supervised methods in both settings and for the unsupervised one in the same-domain setting, only a fraction of the 50 pairs are visible because most of them are too short to be seen due to the alignment.

both embeddings to build a seed dictionary as in "Vecmap identical" (Søgaard, Ruder, and Vulić, 2018), even when those languages are not expected to share vocabulary. With the right self-learning procedure, such initialization can produce an alignment that is systematically competitive with supervised methods.

However, the cross-domain setting raises an additional issue: this chapter has found some evidence that domain-specific vocabulary provokes a lack of global isometry and that the best alignment might be a partial one that focuses on a subset of vocabulary that is common to the domains involved. This thesis advocates for further research in improving the self-learning procedure

of existing unsupervised methods, such that it does not attempt to map the entire vocabularies of both languages involved together. This could help building better cross-domain embeddings in an unsupervised manner, which can be useful when pre-training data is lacking for a specific domain in one language.

But even solving the aforementioned issue might not be sufficient. The results of supervised baselines show that the lack of isometry imposes a glass ceiling on isometry-based methods for some hard-to-align pairs of language and in the cross-domain setting.

The previous chapter has shown that static embeddings might require an explicit cross-lingual signal to provide useful aligned representations, as attempts at language-agnostic static embeddings failed. This chapter has shown that those explicitly aligned static embeddings have important limitations, and the reasons for those limitations are more than just a lack of isometry. This leads the thesis to focus on multilingual contextualized embeddings, rather than static ones, in the following chapters.

Chapter 8

Comparing Multilingual Static and Contextualized Embeddings

The two previous chapters showed that existing unsupervised methods for building multilingual static embeddings have known limitations and do not always provide a good multilingual alignment. On the other hand, existing literature has shown that contextualized embeddings like mBERT can provide nice multilingual properties without any explicit cross-lingual supervision signal. However, those contextualized embeddings are usually evaluated for their cross-lingual transfer abilities rather than for multilingual alignment. And there is a lack of consensus in the literature about whether contextualized embeddings were actually well-aligned.

This chapter proposes an evaluation method for multilingual alignment in contextualized embeddings. Based on a nearest-neighbor search like Bilingual Lexicon Induction (BLI), it allows a direct comparison with static embeddings.

Results show that existing multilingual contextualized embeddings provide a multilingual alignment that is comparable with competitive static embeddings. It also shows that contextualized embeddings provide a stronger form of alignment according to the definition of "strong alignment" provided in Section 2.3.2.

8.1 Proposed Evaluation method for contextualized word-level alignment

This section describes the proposed evaluation methods. It first describes how translated pairs of words are provided with their contexts, it then explains the associated retrieval task.

8.1.1 Using bilingual dictionaries for more exact word alignment

In order to extract translated words from translated sentences with a minimized number of errors, a bilingual dictionary is used.

The proposed evaluation method requires two datasets: a translation dataset containing gold-standard translated pairs of sentences, and a bilingual dictionary. The bilingual dictionary contains pairs of translated words, like the pair "rapide"-"fast" in our example Fig. 8.1. In a pair of translated sentences from the translation dataset, for each word from one sentence, every potential translated word indicated by the bilingual dictionary is collected from the other sentence. A pair of translated words is kept with its context if there is only one candidate for the translation.

The pairs obtained can be seen as a contextualized bilingual dictionary, where the translated words are in accordance with their context. One could not use uncontextualized pairs of words from the original bilingual dictionary directly because the evaluated models are pre-trained on whole

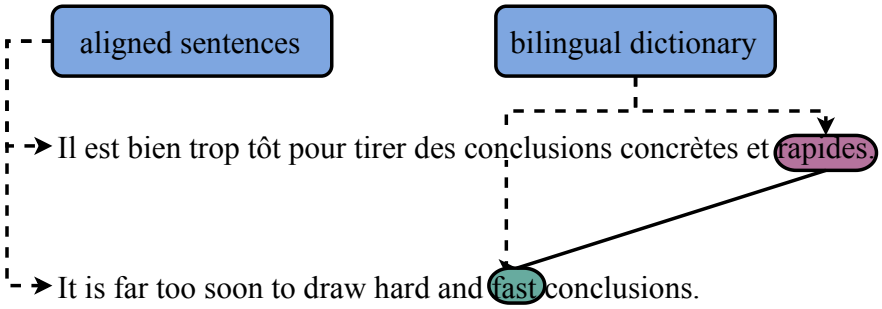


Figure 8.1 – Extracting a contextualized translated pair with a bilingual dictionary.

sentences and, since they have not been pre-trained on single words, they might not give relevant representations of words without context.

Working at the word level instead of the sentence level like Singh et al. (2019) and Pires, Schlinger, and Garrette (2019) is chosen for several reasons. A good alignment of sentence representations does not necessarily guarantee a good alignment of word representations. But more importantly, working at the word level allows a more direct comparison with multilingual word embeddings.

8.1.2 The nearest-neighbor search

Once extracted, the translated-in-context pairs of words are passed through the model we want to evaluate to produce contextualized representations of the translated words.

To build the representation of each contextualized word, the whole sentences are passed to the evaluated model (typically mBERT) and the contextualized representations of the words from the extracted word pairs are kept. Such representations can be extracted for each stacked Transformer block (or layer). By convention, the representation of index 0 will be the one from the input un-contextualized embedding layer. For example, mBERT, which uses 12 Transformer blocks, will produce 13 representations for a word, the 0th one for the initial embedding and the 1st to 12th for the output of each Transformer block.

For each layer of a Transformer-based model, a nearest-neighbor retrieval task is then performed on the produced representations. In a similar manner as Pires, Schlinger, and Garrette (2019), a fixed number N of representations of translated-in-context pairs is randomly sampled. To avoid favoring contextualized models over static embeddings only pairs involving different words are sampled from the entire set of possible evaluation pairs. In other words, the same word is not present twice in the selected set of pairs, since it would lead a static embedding to produce the exact same representation for both.

Then the retrieval task is similar to how BLI is evaluated for static embeddings. The evaluation score is the top-1 accuracy of the nearest-neighbor search for the translation among the N representations.

Following existing approaches for BLI, the similarity between representations is given by the Cross-domain Similarity Local Scaling (CSLS) (cf. Section 2.3.1). It has the advantage of taking into account the density around the compared representations.

8.1.3 Strong and weak alignment

As explained in Section 2.3.2, static embeddings are always evaluated for weak alignment in previous literature. This thesis proposes to also evaluate strong alignment as a stricter evaluation of multilingual alignment.

In practice, the search space for the nearest-neighbor search is simply the set of words from the

source language instead of the target language. More formally, the weak alignment score is given by:

$$S_{\text{weak}}(U, V) = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1} \left[\text{CSLS}(u_i, v_i) > \max_{j \neq i} \text{CSLS}(u_i, v_j) \right] \tag{8.1}$$

Where U and V are the sets of representations of translation pairs (u_i, v_i) and N is the number of sampled pairs. Similarly, the strong alignment score can be written as:

$$S_{\text{strong}}(U, V) = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1} \left[\text{CSLS}(u_i, v_i) > \max_{j \neq i} \text{CSLS}(u_i, \boldsymbol{u}_j) \right] \tag{8.2}$$

The distractors for v_i are all the elements of U (except u_i) instead of V.

8.2 Measuring the multilingual alignment

This section first investigates the fact that different results were obtained regarding sentence-level alignment. Then, it demonstrates that the proposed evaluation method provides better pairs of words than FastAlign thanks to a lesser number of carefully selected pairs. Finally, it shows results for the word-level evaluation for weak alignment first and then for strong alignment, showing that multilingual Transformer-based models bring better alignment than multilingual word embeddings.

8.2.1 Experimental Setup

We compare six models:

mBERT (Devlin et al., 2019) was already detailed in background chapters. It is pre-trained on Wikipedia in the 104 most frequent languages with two objectives: (1) Masked language modeling (MLM); predicting randomly masked out words and (2) next sentence prediction (NSP) determining whether two sentences are consecutive only using the representation of the [CLS] token. It was not trained on any parallel text.

XLM-R (Conneau et al., 2020b) was also described in background chapters. It is pre-trained on CommonCrawl (which contains Wikipedia) on 100 languages. It is based on RoBERTa (Liu et al., 2019b), so it only pre-trains with the MLM objective. It was not trained on any parallel text.

XLM-15 (MLM+TLM) (Conneau and Lample, 2019) pre-trained on Wikipedia in the 15 languages from the XNLI dataset (Conneau et al., 2018a). It is pre-trained on MLM objective but also on parallel text (drawn from XNLI) using the Translated Language Modeling (TLM) objective, which is a MLM objective applied to parallel text, to allow the model to attend to words from the other language to predict masked out words. Additionally to its training on parallel data, its input embeddings is added to a language embedding indicating the language of the sentence, whereas models like mBERT and XLM-R have no input information about the language.

XLM-100 (Conneau and Lample, 2019) pre-trained on Wikipedia in 100 languages with MLM only.

AWESOME (Dou and Neubig, 2021) which is mBERT fine-tuned on a variety of self-supervised objectives and supervised objectives on a parallel corpus to improve word-level alignment for extracting pairs of translated words in parallel sentences: MLM, TLM but also objectives on the consistency of the produced alignment.

mBART (Liu et al., 2020b) contrary to all previously mentioned models which are Transformer encoders, mBART follows an encoder-decoder architecture. It was pre-trained on filling missing spans of texts for 50 languages. We consider only the representations built by the encoder part of the model as we empirically observed that the decoder gives worse multilingual alignment than the encoder.

To evaluate the multilingual alignment of those models, we rely on the WMT19 dataset (Wikimedia Foundation, 2019) for parallel sentences. And For the bilingual dictionaries, we use MUSE (Lample et al., 2018). Monolingual FastText embeddings (Bojanowski et al., 2017) aligned with RCSLS (Joulin et al., 2018) are used as a baseline[16].

For all experiments, the number of sampled pairs is $N = 5000$ as in (Pires, Schlinger, and Garrette, 2019). The number of neighbors for the CSLS criterion is $k = 10$ as in (Joulin et al., 2018). To avoid favoring contextualized models over FastText aligned embedding we chose to sample distinct pairs of words. We empirically verified for all the layers of three models (mBERT, XLM-R, AWESOME) on three language pairs and for 10 different sampling of pairs of words that it gives equivalent results: we observed a strong correlation with a 0.86 Spearman rank correlation (p-value < 0.01). To obtain the 95% confidence intervals on all figures and the empirical standard deviation for all tables, we perform 10 runs of each experiment.

All the code for the experiments in this chapter has been open-sourced and is availabe on Github[17]. Scripts for reproducing the experiments of this chapter can more precisely be found at the following URL:

https://github.com/posos-tech/multilingual-alignment-and-transfer/tree/main/scripts/2022_ijcnn

8.2.2 Comparing different kinds of sentence representations

Before reporting results on word-level alignment, this section investigates the contradiction between Pires, Schlinger, and Garrette (2019) and Singh et al. (2019) on sentence-level alignment for the mBERT model, answering the following open question this thesis has already raised in background chapters:

Open Question 2.2. *How good is the Multilingual Alignment (MA) of raw sentence representations obtained from contextualized word embeddings? What pooling method provides the best alignment?*

To remind the reader, on the one hand, Pires, Schlinger, and Garrette (2019) performed a nearest-neighbor search similar to ours on sentence representations. Each sentence was represented by the average of the embeddings of its tokens and those vectors were centered for each language. High retrieval accuracy is observed with this method for typologically similar languages.

On the other hand, Singh et al. (2019) reported a Canonical Correlation Analysis (CCA) across layers of the representation in various contexts of the initial token [CLS] which is expected to encode the meaning of the sentence. This method shows that those representations are dissimilar, and the dissimilarity grows stronger towards the deeper layers.

In Fig. 8.2 the same decrease is observed across layers, as Singh et al. (2019), for the similarity between [CLS] tokens of translated sentences with the cosine similarity instead of the CCA. But the similarity decreases even more for random pairs of sentences drawn from the same dataset.

The decrease in the similarity between translated pairs can be deduced from the fact that it reaches exactly 1 at the 0th layer, corresponding to non-contextualized embeddings which will have the same value if the token is the same, here "[CLS]". The similarity can do nothing but decrease when information from the context is injected in the contextualized representation of the [CLS] token.

However, to fairly compare representations based on the CLS token and averaged representations of the sub-words, they can both be used in the same nearest-neighbor task as proposed in our method for word representations. The proposed retrieval criterion can be adapted for sentence representations. The sentence representations are not centered as in Pires, Schlinger, and Garrette (2019) in order to evaluate directly the quality of alignment, and not to artificially improve it.

[16] https://fasttext.cc/docs/en/aligned-vectors.html
[17] https://github.com/posos-tech/multilingual-alignment-and-transfer

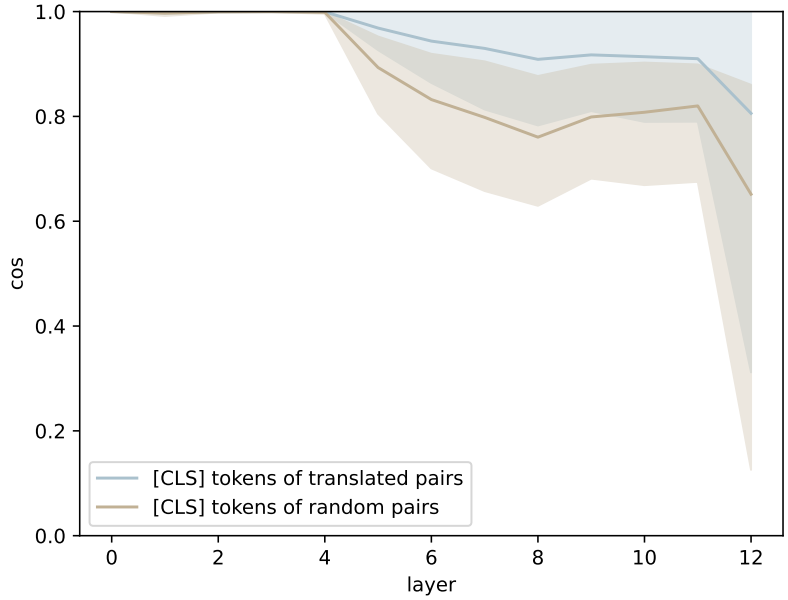


Figure 8.2 – Evolution of the similarity between CLS tokens across layers of mBERT for translated and random pairs (ru-en), with 95% confidence interval.

method and layer	de-en	ru-en	zh-en
FastText avg	53.3 (1.1)	26.1 (1.1)	1.4 (0.2)
CLS first	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)
avg first	56.1 (2.1)	17.8 (0.7)	41.7 (1.1)
CLS best	77.9 (0.4)	59.6 (0.5)	51.1 (0.5)
avg best	**90.1** (0.2)	**82.1** (0.4)	**88.1** (0.6)
CLS last	60.9 (0.8)	40.5 (0.5)	21.0 (0.8)
avg last	87.3 (0.4)	75.5 (0.5)	79.4 (0.3)

Table 8.1 – NN-search for sentence representations of mBERT: first indicates the results for the first layer of mBERT (non-contextualized embedding), last indicates the results for the output layer representations, and "best" is for the layer obtained the higher retrieval accuracy. A baseline with the average of FastText embeddings is provided for comparison.

Results reported in Table 8.1 show that whatever the chosen layer is, averaged representations of the sentences provide better multilingual alignment than CLS representations. For deeper layers, both give better alignment than aligned word embeddings averaged over the sentence. It must also be noted that the alignment of the averaged mBERT representation suffers less from the typological distance between languages than the CLS representation or aligned FastText.

Sentence representations can give different results according to the chosen method, but it seems that whatever it is, sentence representations produced by mBERT are relatively aligned across languages. However, the CLS representation seems to be less relevant. Furthermore, the CLS token does not exist in all Transformer-based models, which are not all trained on a sentence-level pretraining task like NSP for mBERT, hence this thesis recommends relying on averaged representations rather than any special token representation.

8.2.3 Verifying that bilingual dictionary provide more accurate pairs

Fair multilingual alignment of sentence representations does not guarantee a good word-level multilingual alignment. Prior works that looked at word-level alignment of contextualized representations

Table 8.2 – Precision of the extracted pairs

method	en-de	en-fr	ro-en
ours	**90.1**	**95.2**	**94.5**
FastAlign	71.3	80.0	71.8

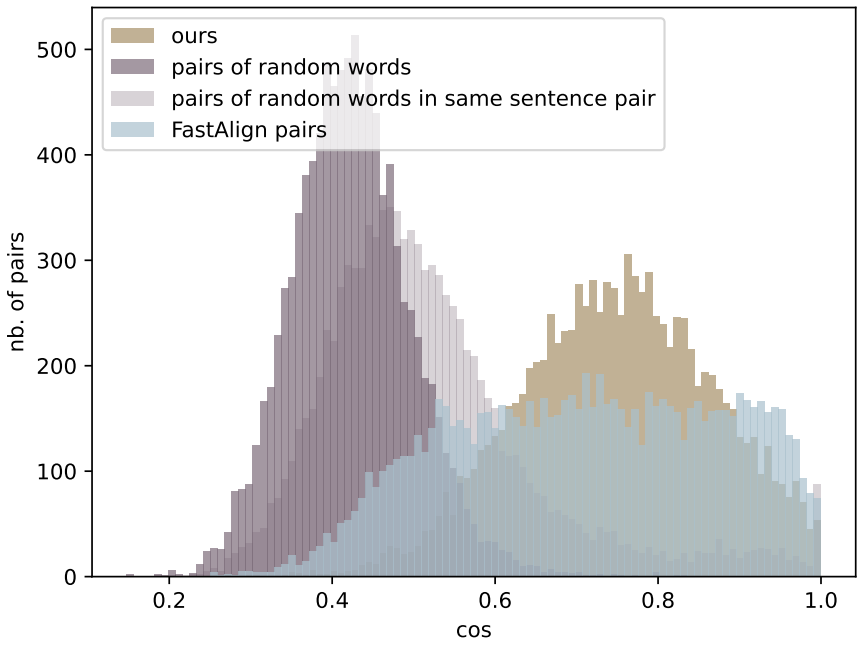


Figure 8.3 – Distribution of similarity between various types of pairs (en-de).

relied on alignment tools like FastAlign (Zhao et al., 2021). However, this thesis hypothesizes that such word alignment tools produce errors that can introduce noise in the evaluation. This section intends to answer the following open question raised earlier on the matter:

Open Question 2.4. *What is the impact of word alignment errors on the measure of multilingual alignment? How can we mitigate the impact of those errors?*

Section 8.1 argued that the proposed method is a way to extract translated words which is less prone to errors.

Table 8.2 shows the proportion of accurate pairs extracted with bilingual dictionaries and FastAlign (Dyer, Chahuneau, and Smith, 2013). It demonstrates that dictionaries allow us to extract more accurate pairs than FastAlign, although it provides fewer pairs in quantity. For 10,000 sentences from WMT19 for the English-German pair: the proposed method extracts 50,590 word pairs and FastAlign 190,665.

Zhao et al. (2021) used FastAlign to compare the similarity of translated pairs of words and random ones with mBERT. When measuring those similarities on the last layer and plotting the sampled distribution, they observe that the pairs obtained with FastAlign give a very broad distribution that overlaps a lot with random pairs, which leads them to conclude that word-level representations built by mBERT are not well aligned across languages and it motivates them to propose a method to re-align representation after pre-training.

But because they use FastAlign, they are considering many unrelated pairs as translations. Figure 8.3 shows those distributions for the eight layer of mBERT. With its right and wrong extracted pairs, the distribution drawn by FastAlign (in blue) overlaps more the distribution of random pairs (in purple) than the distribution of pairs extracted with a bilingual dictionary (in yellow). The distribution of random pairs from the same sentence pair (in light purple) is actually

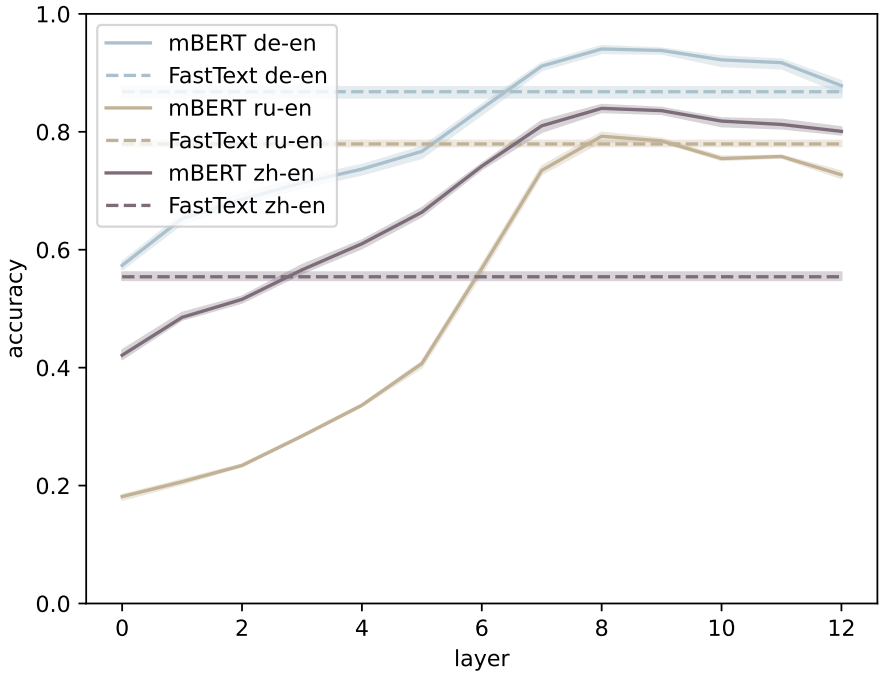


Figure 8.4 – Evolution of S_{weak} across layers for mBERT compared with FastText aligned.

very close to the distribution of random pairs in the whole dataset (in dark purple), which justifies that any error in the extraction of a pair might increase the overlap between extracted pairs and random pairs.

This shows that FastAlign generates too many mistakes to make an accurate evaluation of the multilingual alignment produced by a model.

8.2.4 Contextualized embeddings are well-aligned

Having shown that using a bilingual dictionary is relevant to the issue, this section performs the nearest-neighbor search described in Section 8.1 for evaluating the word-level multilingual alignment of contextualized embeddings, answering the following question from a previous chapter:

Open Question 2.3. *How do contextualized and static multilingual embeddings compare in terms of word-level multilingual alignment?*

Results for mBERT and five language pairs are shown in Figure 8.4. It shows that for a few deep layers, mBERT produces representations that are better aligned than multilingual word embeddings.

Lower layers give worse alignment. The literature also suggests that lower layers encapsulate information about lower-level features like Part-of-Speech tags, while deeper layers may encode higher-level information like meaning (Tenney, Das, and Pavlick, 2019). This suggests that multilingual patterns are linked to high-level features. It also goes against, but does not surely invalidate, the hypothesis made by several papers (Pires, Schlinger, and Garrette, 2019; Singh et al., 2019; Wu and Dredze, 2019), that shared vocabulary is what allows models like mBERT to align representations without having been exposed to parallel texts.

The very last layers also give worse results than layers 8 to 10. Wu and Dredze (2019) have shown that mBERT representations hold language-specific information at each layer. These language-specific components might take more importance on the last layers as the pre-training objective is

layer	model	de-en	ru-en	zh-en
-	FastText	86.8 (0.67)	77.9 (0.39)	55.4 (0.43)
best	mBERT	**94.1** (0.47)	79.2 (0.64)	**84.0** (0.40)
	XLM-100	83.5 (0.52)	67.4 (0.38)	27.4 (0.39)
	XLM-R Base	87.7 (0.41)	68.7 (0.24)	63.6 (0.50)
	XLM-R Large	88.9 (0.30)	76.8 (0.15)	72.3 (0.36)
	XLM-15[a,b]	68.5 (0.31)	28.4 (0.30)	24.5 (0.56)
	AWESOME[a]	93.4 (0.53)	76.1 (0.60)	82.7 (0.38)
	mBART[b,c]	92.1 (0.52)	**81.2** (0.79)	74.7 (0.34)
last	mBERT	87.9 (0.74)	72.7 (0.50)	80.1 (0.43)
	XLM-100	82.4 (0.60)	64.6 (0.43)	25.7 (0.46)
	XLM-R Base	77.2 (0.81)	49.3 (0.68)	51.0 (0.35)
	XLM-R Large	77.4 (0.49)	53.5 (0.65)	54.4 (0.43)
	XLM-15[a,b]	28.5 (0.60)	4.6 (0.23)	11.2 (0.39)
	AWESOME[a]	86.5 (0.47)	67.4 (0.21)	79.4 (0.44)
	mBART[b,c]	92.1 (0.52)	81.2 (0.79)	74.7 (0.34)

[a] uses parallel data in pre-training
[b] encodes the language in input
[c] encoder-decoder model (we only evaluate the encoder)

Table 8.3 – NN-search results for more models

to predict masked words, a language-specific task. Indeed, the model must learn not to replace a masked-out word with its translation.

This type of curve for S_{weak} is observed for all the multilingual models we evaluated. Table 8.3 shows retrieval scores for the first, best, and last layers of different models.

The best layer of mBERT gives the best results with respect to all other models. It might be due to the fact that the next sentence prediction task helps, an objective on which it is the only pre-trained model. However, it could also be explained by the fact that mBERT is trained solely on Wikipedia whereas models like XLM-R are trained on the CommonCrawl corpus which might contain texts that are less comparable across languages. It is also to be noted that the TLM objective on parallel texts proposed by the XLM model seems to make the alignment worse.

As the different evaluated models have many differences and obtain somewhat similar results, one cannot isolate a single parameter that makes the multilingual alignment better or worse. Nevertheless, it seems that most of those multilingual models build multilingual representations that are competitive with word embeddings that have been explicitly aligned.

8.2.5 Contextualized embeddings are strongly aligned

Finally, the same models are evaluated for the strong alignment retrieval criterion S_{strong} defined in Equation 8.2, allowing to answer the last open question regarding the gaps in the evaluation of multilingual alignment:

Open Question 2.1. *Are multilingual static embeddings competitive in terms of strong alignment?*

Results for mBERT are reported in Figure 8.5 and results for all models are reported in Table 8.4. They show that language-agnostic contextualized embeddings provide a significantly stronger alignment than static embeddings.

The multilingual alignment of most models seems to be robust. Retrieval accuracy is significantly greater for those multilingual models than for multilingual word embeddings. There is yet again no way to identify what makes one Transformer-based model perform better than another. Nevertheless, the results reported in this chapter demonstrate that there is a strong word-level alignment in most multilingual Transformer-based language models, even for language-agnostic models like mBERT and XLM-R.

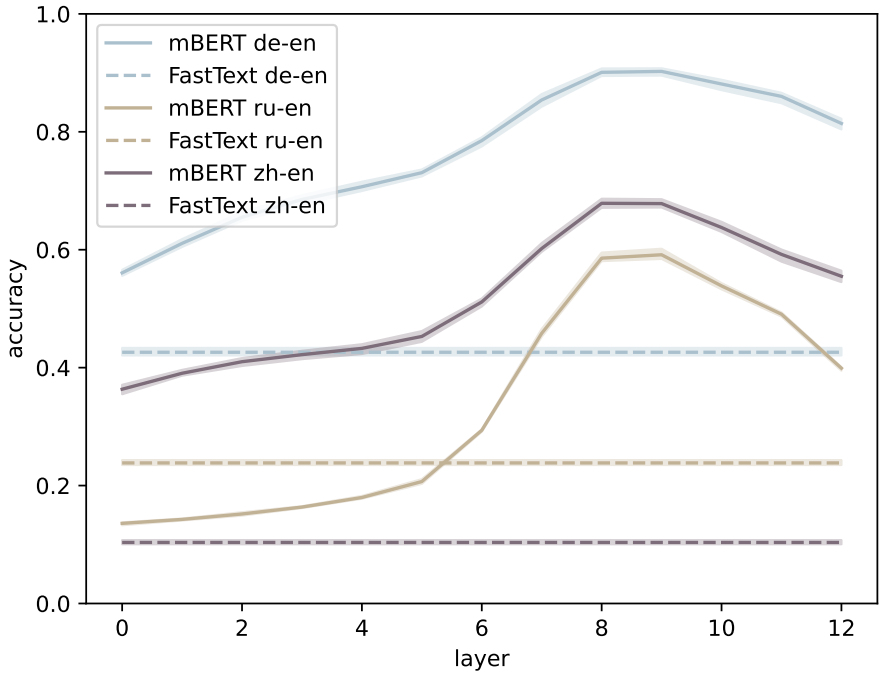


Figure 8.5 – mBERT and FastText aligned representation evaluated with S_{strong}

layer	model	de-en	ru-en	zh-en
-	FastText	42.6 (0.47)	23.8 (0.33)	10.4 (0.23)
best	mBERT	90.3 (0.47)	59.1 (0.68)	67.9 (0.49)
	XLM-100	81.2 (0.49)	57.9 (0.58)	22.5 (0.66)
	XLM-R Base	82.9 (0.46)	53.5 (0.69)	49.7 (0.45)
	XLM-R Large	87.6 (0.40)	**70.4** (0.61)	65.1 (0.47)
	XLM-15[a,b]	62.3 (0.63)	16.7 (0.28)	21.6 (0.58)
	AWESOME[a]	**91.6** (0.82)	64.6 (0.63)	**70.9** (0.40)
	mBART[b,c]	88.4 (0.45)	68.0 (0.82)	60.0 (0.58)
last	mBERT	81.5 (0.63)	39.9 (0.27)	55.5 (0.65)
	XLM-100	72.9 (0.37)	39.0 (0.58)	18.6 (0.55)
	XLM-R Base	73.6 (0.49)	36.4 (0.65)	38.6 (0.39)
	XLM-R Large	72.2 (0.41)	40.5 (0.50)	42.6 (0.48)
	XLM-15[a,b]	20.2 (0.79)	3.7 (0.20)	8.4 (0.48)
	AWESOME[a]	80.2 (0.77)	40.5 (0.18)	56.9 (0.33)
	mBART[b,c]	88.4 (0.45)	68.0 (0.82)	60.0 (0.58)

[a] uses parallel data in pre-training
[b] encodes the language in input
[c] encoder-decoder model (we only evaluate the encoder)

Table 8.4 – NN-search results for strong alignment

8.3 Conclusion

This chapter shows that multilingual contextualized embeddings, even when trained without any explicit cross-lingual signal, can provide good multilingual alignment. It shows that mean-pooling

representation is a better way to build multilingual sentence representations.

More importantly, it shows that alignment in static embeddings can be brittle, as they lack strong alignment, contrary to contextualized embeddings. Having found that contextualized embeddings are well-aligned, it raises the question whether there is a link between this alignment and the cross-lingual transfer abilities of models like mBERT and XLM-R, as discussed in the next chapter.

Chapter 9

The link between Multilingual Alignment and Cross-lingual Transfer

The previous chapter has shown that, on top of their Cross-lingual Transfer Learning (CTL) abilities, models like mBERT provide contextualized representations that have a good multilingual alignment. On the other hand, the literature indicates that improving alignment with realignment methods does not systematically improve cross-lingual transfer.

The current chapter thus studies the link between alignment and cross-lingual transfer. It also evaluates the impact of fine-tuning on alignment, showing that, depending on the fine-tuning task, there isn't necessarily a catastrophic forgetting of multilingual properties, even though fine-tuning is performed in a single language. Finally, this sheds light on the relative failure of realignment methods, identifying conditions for their success.

9.1 A strong correlation between alignment and cross-lingual Transfer

For all experiments in this chapter, four language models of varying sizes are used to exemplify the effect of scale on the results: XLM-R, which at least exists with two sizes (Base and Large), and mBERT, which is smaller than XLM-R Base and has an even smaller version that is obtained through distillation: distilmBERT.

9.1.1 How to evaluate CTL

A model has high CTL abilities when, after fine-tuning for one language, it can obtain a high evaluation score in other languages. To evaluate it for a given task, this thesis proposes to compute the relative difference between the evaluation metric m_{en} on the English test set and the evaluation metric m_{tgt} on the target language:

$$\text{cross-lingual transfer} = \frac{m_{\text{tgt}} - m_{\text{en}}}{m_{\text{en}}} \tag{9.1}$$

The monolingual metric is a score between 0 and 1, like accuracy or f1-score, where higher is better. The CTL metric gives scores between -1 and $+\infty$. A negative score is obtained if and only if $m_{\text{tgt}} < m_{\text{en}}$, which should always be the case in practice. Values closer to 0 then indicate better CTL for a specific task and language.

It must be noted that for datasets where the target language test set is a translation of the English one, the normalization in Equation 9.1 allows the metric to boil down roughly to minus the proportion of correct answers in English that were misclassified when translated, assuming there aren't too many misclassified English examples that were correctly classified in the target language,

	POS	NER	XNLI
en-train	12,543	20,000	392,703
en-dev	2,002	10,000	2,490
en-test	2,077	10,000	5,010
ar-test	680	10,000	5,010
es-test	426	10,000	5,010
fr-test	416	10,000	5,010
ru-test	601	10,000	5,010
zh-test	500	10,000	5,010

Table 9.1 – Number of examples

which should be the case since there are not that much misclassified English examples in general, although it might depend on the task.

9.1.2 How to evaluate multilingual alignment

To evaluate multilingual alignment, the same nearest-neighbor criterion is used as the previous chapter in Section 8.1, with a few differences. The most important difference is that since the evaluation method is not used here for comparison with static embeddings, this chapter allows to sample different pairs of contextualized words that involve the same words, which makes the task harder. Also, simple cosine similarity is used, instead of the CSLS criterion, to avoid having additional confounding factors in the measure of correlation between alignment and CTL. Finally, more pairs are sampled (10,000 instead of 5,000) to make the task more difficult and thus obtain more diverse results across models.

As in the previous chapter, the evaluation is performed for weak alignment as well as strong alignment. The evaluation could be done with other ways to extract related pairs than a bilingual dictionary, using word alignment tools, like FastAlign and a more accurate one: AWESOME-align (Dou and Neubig, 2021). However, in practice, conclusions were unchanged when using different word alignment tools.

9.1.3 Experimentals details

9.1.3.1 Fine-tuning for cross-lingual transfer

We evaluate cross-lingual transfer with three multilingual tasks, the sizes of which are reported in Table 9.1:

- Part-of-speech tagging (POS-tagging) with the Universal Dependencies dataset (Zeman et al., 2020). Similarly to Wu and Dredze (2020), we use the following treebanks: Arabic-PADT, English-EWT, Spanish-GSD, French-GSD, Russian-GSD, and Chinese-GSD.
- Named Entity Recognition (NER) with the WikiANN dataset (Pan et al., 2017).
- Natural Language Inference (NLI) with the XNLI dataset (Conneau et al., 2018b).

It must be noted that XNLI is the only dataset with translated test sets, and thus the only one for which the cross-lingual transfer metric is strictly comparable across languages. A high correlation will nonetheless be observed between CTL and alignment for the two other tasks, suggesting that the CTL metrics is not so much affected by differences in size and domain between the test sets.

Fine-tuning was done following the experimental setup from Wu and Dredze (2020). Adam is used, with a learning rate of `2e-5` with a linear decay and warmup for 10% of the steps. Fine-tuning is performed on 5 epochs, and 32 batch size, except for XNLI in the second experiment, where we trained for 2 epochs, which still leads to more fine-tuning steps than any of the two other tasks (cf. Table 9.1).

task	layer	weak		strong	
		before	after	before	after
POS	last	0.58	0.84	0.82	0.87
	penult	0.78	0.84	0.87	0.86
NER	last	0.69	0.72	0.86	0.70
	penult	0.82	0.71	0.87	0.82
NLI	last	0.51	0.75	0.86	0.82
	penult	0.74	0.95	0.84	0.92

Table 9.2 – Spearman's rank correlation of CTL with the English-target alignment produced by the last and penultimate layer before and after fine-tuning. Evaluation is done across 5 languages, 5 seeds, and 4 models ($N = 100$). All cells have p-value < 0.05.

For the realignment methods, still following Wu and Dredze (2020), training is done in in a multilingual fashion, where each batch contains examples from all target languages. However, the same learning rate and schedule as for fine-tuning are used for a fair comparison between joint and sequential realignment, since the same optimizer is used for fine-tuning and realignment when performing joint realignment. The maximum sequence length was 96 like Wu and Dredze (2020) but a batch size was 16 instead of 128 due to limited computing resources.

From a translation dataset, realignment pairs were extracted either using a bilingual dictionary, following the method introduced in the previous chapter, or either with FastAlign Dyer, Chahuneau, and Smith, 2013 or AWESOME-align Dou and Neubig, 2021. For FastAlign, alignments were produced in both directions and symmetrized with the `grow-diag-final-and` heuristic provided with FastAlign, following the setting of Wu and Dredze (2020). For all methods of extraction, only one-to-one alignments were kept and trivial cases where both words are identical were discarded, again following Wu and Dredze (2020).

9.1.4 Measuring the correlation

This section studies the correlation between CTL and multilingual alignment. Alignment is measured in different manners, with weak and strong alignment, and before and after fine-tuning on a specific task in English.

Table 9.2 reports those correlation values obtained from 100 samples each, with four different models (distilmBERT, mBERT, XLM-R Base, and Large), five target languages (Arabic, Spanish, French, Russian and Chinese) and five seeds for initialization of the classification head and shuffling of the fine-tuning data.

Results show that strong alignment is better correlated to cross-lingual transfer than weak alignment. With the exception of two tasks after fine-tuning (NER and NLI), strong alignment has a marginally higher correlation with CTL. This is particularly noticeable when looking at alignment before fine-tuning on the last layer, going from a correlation between 0.51 and 0.69 for weak alignment to one ranging from 0.82 to 0.86 for strong alignment.

Table 9.2 also shows that for NLI, the alignment on the penultimate layer seems better correlated to cross-lingual transfer than with the last layer. A relatively important gap in correlation is measured between the last and the second-before-last layer for all cases except for strong alignment before fine-tuning. The fact that alignment on the penultimate layer would correlate better than the last for NLI can be explained by the sentence-level nature of the task. For sentence classification tasks, the classification head reads only the representation of the first token of the last layer, which is computed from the representations of all the tokens at the previous layer, leading to a pooling of the penultimate layer.

Despite the different values observed, there seems to be no significant difference between correlation for alignment measured before and after fine-tuning, and a careful analysis of the confidence interval obtained with bootstrapping (Efron and Tibshirani, 1994) can confirm this (cf. Appendix 2 for detailed results).

Figure 9.1 shows one of the cases with the highest correlation (0.92). The correlation seems to

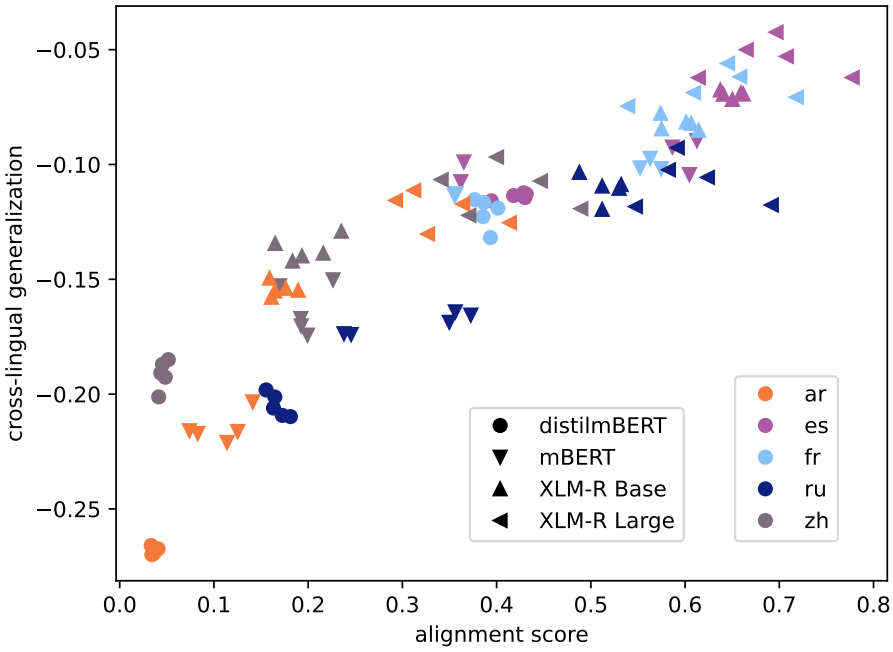


Figure 9.1 – Plot of CTL abilities against the English-target strong alignment measured for the penultimate layer after fine-tuning on NLI.

hold well across models (forms) and languages (colors). However, for a given model and language, the random seed for fine-tuning seems to be detrimental to the correlation, although at a small scale. Hence, alignment might not be the only factor to affect cross-lingual generalization as the model initialization or the data shuffling seems to play a smaller role.

For two of the three tested tasks (NER and POS-tagging), it must be noted that the CTL metric is not strictly comparable across languages since the test sets for each language are of different domains and sizes. However, for the third task (NLI), each test set is a translation of the English one, and thus the CTL metric is strictly comparable in that case. This might explain why correlations are higher for the NLI task than the others. Nevertheless, the observed correlation for the two other tasks is still significantly high, which suggests that the general tendency might not be affected by the differences in domains and sizes in the test sets.

This section has shown that there is an undeniably strong correlation between multilingual alignment and cross-lingual transfer, answering one of the most central open question of this thesis:

Open Question 4.2. *Is multilingual alignment correlated with cross-lingual transfer?*

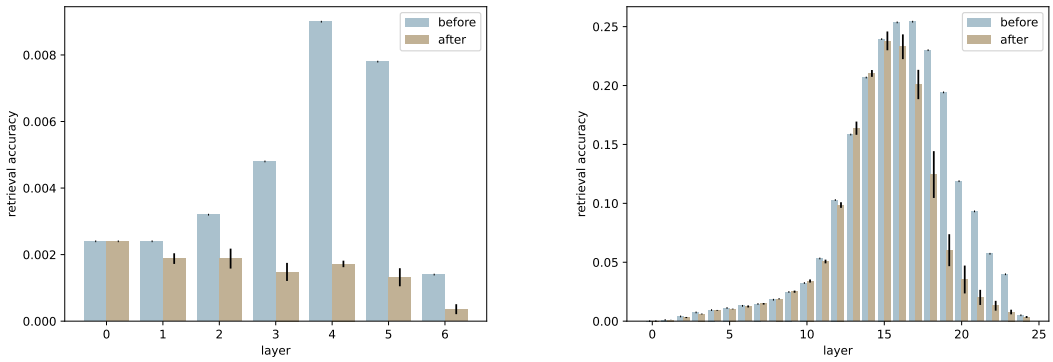
However, if alignment is linked with CTL, it is then all the more surprising that the literature shows that realignment methods do not significantly improve CTL abilities. But before trying to identify the reasons for this apparent contradiction, the next section studies the impact of fine-tuning on alignment, showing that different tasks have different effects on alignment.

9.2 The impact of fine-tuning on alignment

The previous section has already shown that strong alignment is highly correlated with CTL. However, results did not allow us to conclude about whether alignment measured before or after fine-tuning was better correlated to CTL abilities. To understand the difference between both measures, this section studies the impact of fine-tuning on the alignment of mLMs representations, using the same fine-tuning runs as in the previous section.

task	model	en-ar	en-es	en-fr	en-ru	en-zh
POS	distilmBERT	-0.74	-0.86	-0.87	-0.87	-0.96
	mBERT	-0.90	-0.86	-0.93	-0.95	-0.96
	XLM-R Base	-0.43	-0.46	-0.46	-0.70	0.69
	XLM-R Large	-0.30	0.23	0.44	-0.44	0.26
NER	distilmBERT	0.00	-0.61	-0.60	-0.33	0.00
	mBERT	-0.28	-0.36	-0.49	-0.27	-0.25
	XLM-R Base	5.88	0.22	0.62	1.32	21.99
	XLM-R Large	16.34	2.22	3.17	3.10	12.67
NLI	distilmBERT	5.49	0.30	0.88	2.28	9.78
	mBERT	5.65	0.99	1.08	1.45	5.85
	XLM-R Base	11.17	1.01	1.55	2.67	27.58
	XLM-R Large	25.36	1.78	2.96	2.99	13.57

Table 9.3 – Relative variation of strong alignment at the last layer before and after fine-tuning for different fine-tuning tasks. "$\pm$" indicates standard deviation.



(a) distilmBERT POS

(b) XLM-R Large POS

Figure 9.2 – Evolution across layers of English-Arabic strong alignment before and after fine-tuning of distilmBERT and XLM-R Large on POS-tagging, starting at 0 for the embedding layer.

Table 9.3 shows the relative variation in alignment before and after fine-tuning for all tasks and models tested. The relative difference is built in the same way as the cross-lingual transfer evaluation (Eq. 9.1). Negative values indicate a drop in alignment. Alignment is measured at the last layer. Figures 9.2 and 9.3 show a breakdown by layer for a few cases.

For certain combinations of models and tasks, fine-tuning is detrimental to multilingual alignment. distilmBERT and mBERT mainly show a decrease in alignment for POS-tagging and NER and smaller improvements than other models on NLI. However, POS-tagging is the only of the three tasks that shows dramatic drops where alignment can be reduced by as much as 96%.

The drop in alignment can be explained by catastrophic forgetting. If the model is only trained on a monolingual task, it might not retain information about other languages or about the link between English and other languages.

What is more surprising is the increase in alignment obtained in other cases. XLM-R Base and Large, which are larger models than mBERT and distilmBERT, have a relative increase that can go as high as 25.36 on the NLI task for distant languages. Although these increases are from a small alignment measure, a large increase can still be observed for middle layers where the initial alignment is already quite high (cf. Figure 9.3).

The alignment of larger models being less harmed by fine-tuning is coherent with the fact that

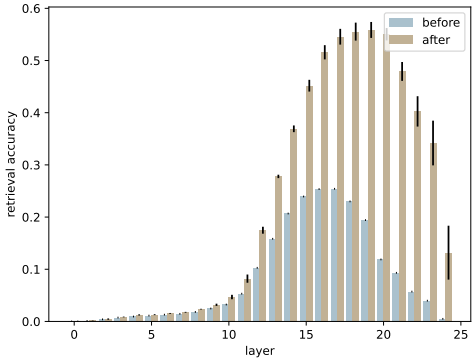


Figure 9.3 – Strong alignment measured across layers of English-Arabic before and after fine-tuning of XLM-R Large on NLI.

those same larger models have been shown to have better CTL abilities. Figure 9.2 shows that more layers seem to mitigate the potentially negative impact of fine-tuning on alignment, as it affects mainly the layers closest to the last one and as the initial alignment measure is globally higher for XLM-R than distilmBERT (before fine-tuning: ≈0.25 against ≈0.008).

Giving a definitive answer as to why different tasks have different impacts on alignment might need further research. However, one could already argue that each task corresponds to different levels of abstraction in NLP. Tasks with a low level of abstraction like POS-tagging might rely on the word itself and thus on more language-specific components of the representations, which when enhanced, decreases alignment. On the other hand, NLI has a higher level of abstraction, requiring the meaning rather than the word form, which might be encoded in deeper layers (Tenney, Das, and Pavlick, 2019) which are more aligned.

Fine-tuning mLMs on a downstream task has an impact on the multilingual alignment of the representations produced by the model. For "smaller" language models, it is systematically detrimental, as well as for certain tasks like POS-tagging. This might explain why some realignment methods might not work for all models nor all tasks (Wu and Dredze, 2020).

9.3 Identifying conditions for the success of realignment methods

This section demonstrates the importance of two factors for the success of a given realignment method:

- The quality of the pairs extracted (with a bilingual dictionary rather than a probabilistic word alignment tool)
- whether alignment already increases when fine-tuning the given model on the given task, as seen in the previous section.

This section uses the realignment method from Wu and Dredze (2020) with strong alignment that was already described in Section 4.3.2.1, Equation 4.8, and introduces two possible modifications for a controlled experiment:

- using different ways to extract word pairs: with a bilingual dictionary, AWESOME-align (Dou and Neubig, 2021), or the originally used word-aligner: FastAlign
- to allow performing realignment jointly with fine-tuning, as described in the following Section.

9.3.1 Joint realignment

Sequential realignment is the usual way to perform realignment: realignment steps are performed on the pre-trained model before fine-tuning. This assumes that the alignment before fine-tuning is positively linked to the cross-lingual transfer abilities of the model and that improving alignment before fine-tuning will improve transfer. However, previous sections and the literature (Efimov et al., 2023) have shown that fine-tuning itself can have an impact on alignment. This section proposes to compare it with joint alignment, where one optimizes simultaneously for the realignment and the downstream task, to try and identify whether alignment before or after fine-tuning is more strongly related to CTL.

The joint realignment approach optimizes simultaneously for a realignment loss and the fine-tuning loss. In practice, for each optimization step, it computes the loss $\mathcal{L}_{\text{task}}$ for a batch of the fine-tuning task and the loss $\mathcal{L}_{\text{realign}}$ for a batch of the alignment data. The total loss for each backward pass is then written as:

$$\mathcal{L} = \mathcal{L}_{\text{task}} + \mathcal{L}_{\text{realign}} \tag{9.2}$$

This joint realignment can be framed as multi-task learning. The fine-tuning task would be the main task and the realignment task an auxiliary one. There are more elaborate methods for training a model with an auxiliary task (Du et al., 2018; Liebel and Körner, 2018; Liu et al., 2019a; Zhang, Duh, and Van Durme, 2018) but the aim here is to propose the simplest method possible to compare joint and sequential alignment in a controlled setting.

9.3.2 Experimental settings

In the same settings as the previous experiments (tasks, models and languages, and the number of seeds), models were fine-tuned in English with different realignment methods and CTL was evaluated in the same languages. Following a similar setting as Wu and Dredze (2020), realignment data from the five pairs of languages (English-target) is interleaved to form a single multilingual realignment dataset. Models are fine-tuned on POS-tagging or NER for five epochs and 2 epochs for NLI because its training data is larger. We use the opus100 translation dataset (Zhang et al., 2020) from which we extract pairs of words using bilingual dictionaries.

9.3.3 The impact of the choice of the model and the task

Results are reported in Table 9.4. It shows that realignment methods improve performance only on certain tasks, models, and language pairs.

Realignment methods, either sequential or joint, provide significant improvement for all models for the POS-tagging task, but less significant ones for NER, and no significant improvement for NLI. The positive impact of realignment on cross-lingual transfer seems to be mirrored by the negative impact of fine-tuning over alignment. Indeed, POS-tagging is also the task for which fine-tuning is the most detrimental to multilingual alignment, as shown in the previous section.

The same parallel can be drawn for models. distilmBERT is the model that benefits the most from realignment. It is also the one whose alignment suffers the most from fine-tuning. Smaller multilingual models seem to benefit more from realignment, as well as they see their multilingual alignment reduced after fine-tuning. In the same way that fine-tuning mainly affects the deeper layers, it is possible that realignment might affect only those deeper layers. This would mean that most layers would have their alignment significantly improved for small models like distilmBERT (6 layers), while larger models might be only superficially realigned.

Finally, besides tasks and models, it can also be observed that the impact of realignment varies across language pairs. Although the experiments were not performed on many language pairs, results are coherent with the idea that realignment methods tend to work better on distant pairs of languages (Kulshreshtha, Redondo Garcia, and Chang, 2020).

On a side note, our controlled experiment does not allow us to conclude whether it is more important to improve alignment before fine-tuning or after. Indeed, there is no clear difference

	en	ar	es	fr	ru	zh	avg
POS-tagging							
distilmBERT	***96.1***	*51.0*	*84.1*	*85.3*	*81.2*	*64.1*	*73.1*
+ before	96.1	65.5	85.8	**86.5**	**84.7**	**67.4**	**78.0**
+ joint	96.1	**66.8**	**85.8**	86.5	84.1	66.4	77.9
mBERT	***96.7***	*51.7*	*85.6*	*86.0*	*82.1*	*66.0*	*74.3*
+ before	96.6	65.1	**86.2**	**86.9**	**84.4**	**68.3**	78.2
+ joint	96.6	**66.5**	86.1	86.9	83.9	68.1	**78.3**
XLM-R base	*95.9*	*62.5*	*86.6*	*86.9*	***86.9***	*70.9*	*78.8*
+ before	96.0	**67.3**	**86.9**	**87.3**	86.8	**71.2**	**79.9**
+ joint	95.9	66.6	86.6	87.2	86.0	70.6	79.4
XLM-R large	***97.7***	***65.1***	***87.0***	***87.5***	***87.0***	***71.5***	*79.6*
Named Entity Recognition							
distilmBERT	*82.9*	*34.5*	*69.2*	*76.1*	*60.2*	*46.8*	*57.4*
+ before	82.9	41.6	67.9	76.4	60.3	48.3	58.9
+ joint	83.0	**42.2**	69.5	76.6	61.2	48.8	**59.6**
mBERT	*84.4*	*40.7*	*74.3*	*79.9*	*63.9*	*52.1*	*62.2*
+ before	84.3	42.1	73.4	80.1	64.9	52.8	62.7
+ joint	84.2	**46.0**	**76.6**	**81.1**	65.5	**54.9**	**64.8**
XLM-R base	*80.0*	*46.4*	*71.8*	*75.0*	*61.6*	*47.4*	*60.4*
+ before	80.0	**55.8**	**76.9**	**77.3**	62.0	47.5	**63.9**
+ joint	79.9	50.3	75.2	75.9	**63.6**	50.1	63.0
XLM-R large	***83.8***	***45.1***	***75.6***	***80.7***	***70.5***	***53.0***	*65.0*
Natural Language Inference							
distilmBERT	*76.0*	*58.2*	*68.5*	***68.7***	*62.3*	*63.4*	*64.2*
+ before	**76.2**	58.4	69.2	68.0	62.6	63.1	64.3
+ joint	76.2	**59.8**	**69.2**	68.6	**63.0**	**64.4**	**65.0**
mBERT	***80.2***	*65.2*	*73.8*	***72.9***	*68.3*	*68.5*	*69.7*
+ before	79.0	63.2	72.9	71.7	66.9	68.7	68.7
+ joint	79.9	**65.6**	**73.8**	72.5	**68.8**	**69.0**	**70.0**
XLM-R base	*82.8*	*70.1*	*77.4*	*76.5*	*74.2*	*71.7*	*74.0*
+ before	81.2	68.4	76.0	74.9	72.8	71.4	72.7
+ joint	**83.7**	**70.8**	**78.0**	**76.7**	**74.6**	**72.7**	**74.6**
XLM-R large	***87.9***	***77.5***	***83.2***	***81.9***	***79.1***	***78.2***	*80.0*

Table 9.4 – Impact of realignment on different models, for various tasks and various target languages. Results are the average over five seeds. The standard deviation is not shown for readability, but a light gray color indicates an increase or decrease that is within the standard deviation and dark gray indicates a decrease that is below one standard deviation. Non-colored cells are either non-realigned baselines or cases where realignment provides an increase above one standard deviation.

	POS	NER
XLM-R base	*78.8*	*60.4*
+ before fastalign	78.6	61.4
+ before awesome	78.6	62.0
+ before dico	**79.9**	**63.9**
+ joint fastalign	78.0	62.1
+ joint awesome	77.8	62.3
+ joint dico	79.4	63.0

Table 9.5 – Average CTL abilities for XLM-R with different types of realignment.

between joint and sequential realignment. It seems that the alignment measured before and the one measured after fine-tuning are equally important to cross-lingual transfer.

Realignment methods unsurprisingly provide better results when the alignment is lower, be it before or after fine-tuning. Distant languages and small models have lower alignment, and POS-tagging is a task where alignment decreases after fine-tuning. Realignment helps only up to a certain point where representations are already well aligned, and CTL gives already good results. For distilmBERT on POS-tagging for transfer from English to Arabic, it provides a +15.8 improvement over baseline, even outperforming XLM-R Large by 1.7 points. In such conditions, realignment is an interesting alternative to scaling for multilingual models.

9.3.4 The impact of the word pair extraction

If realignment succeeds in some favorable conditions, then how can we explain that realignment methods were shown to not be significantly improving CTL on several tasks, including POS-tagging (Wu and Dredze, 2020)? Firstly, to the best of my knowledge, realignment was never tried on distilmBERT or other models of equivalent size. Secondly, Table 9.5 shows that it might be partly due to an element of the realignment methods that was overlooked: the source of related pairs of words.

The way pairs are extracted seems to be crucial to the success of realignment methods. Table 9.5 shows the effect of different types of pair extraction in realignment methods. Realignment methods using pairs extracted with FastAlign or AWESOME-align do not provide significant improvements over the baseline, whereas using a bilingual dictionary does. Using a bilingual dictionary is more accurate for extracting translated pairs as shown in the previous chapter. But another explanation could be that the type of words contained in a dictionary might help since it might contain more lexical words holding meaning and fewer grammatical words.

9.4 Conclusion

This chapter demonstrates that the CTL ability of a multilingual model is highly correlated with the strong alignment of its inner representations.

However, the literature also shows, and it is confirmed by the experiments of this chapter, that improving the alignment of such models does not systematically provide improved cross-lingual generalization abilities. This chapter identifies two main reasons for the failure of realignment methods.

The first cause of failure for realignment is the use of error-prone word alignment tools like FastAlign. This chapter solves this issue by proposing the same extraction based on bilingual dictionaries as proposed in the previous chapter.

The second cause of failure is when the given model already sees its multilingual alignment improve when fine-tuned on the given task. It is as if the given model was already sufficiently well-aligned for the given task such that further realignment would not bring anything. This chapter proposed joint realignment as a potential solution, but the results do not show significant

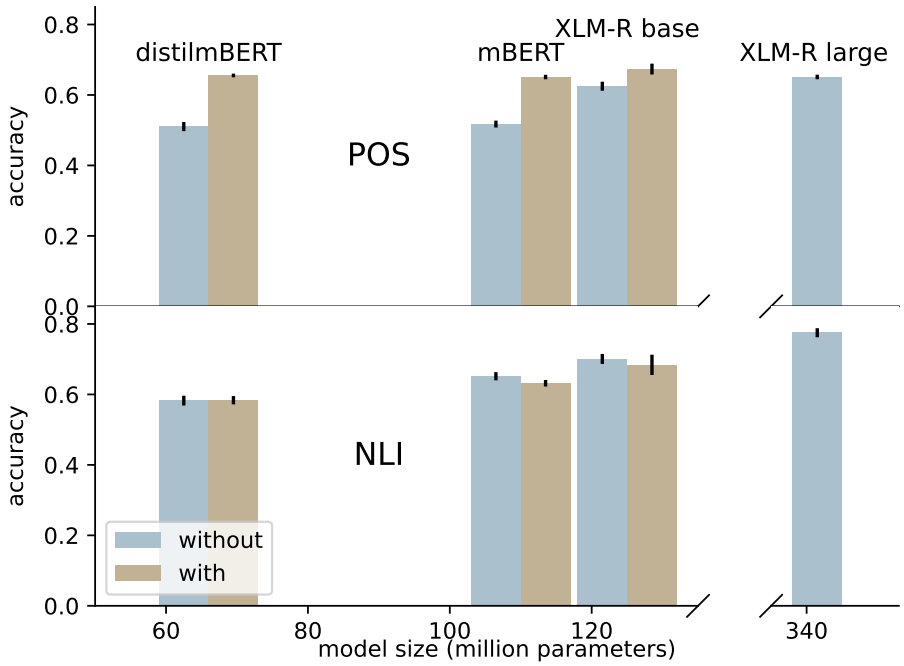


Figure 9.4 – Cross-lingual transfer between English and Arabic with and without realignment, using a bilingual dictionary. For some tasks, realignment can make small models competitive with a large baseline.

improvement over sequential realignment. It might be that further realignment is useless in those cases and that something else would be needed to improve results.

Nonetheless, this chapter finds that realignment works well for low-level tasks and small multilingual models, allowing smaller models like distilmBERT (66M parameters) to be competitive with XLM-R Large (354M parameters) as shown on Figure 9.4.

Chapter 10

Comparing cross-lingual transfer and translation for a clinical NER task

Previous chapters have shown the limitations of static multilingual embeddings while emphasizing the opportunities brought by contextualized models. This chapter compares those contextualized models with simple translation-based baselines. While this was already done in the general domain in the literature (Yarmohammadi et al., 2021), this chapter focuses on the clinical domain, where domain-specific language models and translation models can be used. This chapter eventually answers the following open question raised earlier:

Open Question 5.1. *How do translation-based methods and cross-lingual transfer compare in the clinical domain? Is it possible to leverage domain-specific language models with translation-based methods?*

Because of the lack of clinical multilingual annotated datasets, this chapter focuses on Named Entity Recognition (NER). It proposes a small French test dataset based on the English dataset n2c2 (Henry et al., 2019). It is annotated with the same guidelines to allow cross-lingual transfer. This chapter first describes how the dataset was collected before detailing the results of the comparison of CTL and translation-based methods.

10.1 A new clinical dataset: MedNERF

The CTL abilities of mLMs can be assessed on several tasks in many languages thanks to multilingual benchmarks like XTREME (Hu et al., 2020a) which cannot be used to evaluate medical NER models since they contain only general-domain tasks. Despite the existence of non-English medical NER datasets like QUAERO in French (Névéol et al., 2014) or GGPONC in German (Borchert et al., 2022), CTL cannot be evaluated on these datasets as there is no English counterpart annotated with the same guidelines, which is a pre-requisite for CTL evaluation. In order to tackle this issue, Frei, Frei-Stuber, and Kramer (2022) have introduced a small test dataset of 30 German medical sentences from Electronic Health Records (EHR) annotated in the same way as the n2c2 dataset and have used it to assess the performances of their GERNERMED++ model. Following their path, this thesis proposes to release a medical NER dataset in French based on drug prescriptions.

During the making of this thesis, MedNERF[18] was released. It is a Medical NER dataset in the French language. It was built using a sample of French medical prescriptions annotated with the same guidelines as the n2c2 dataset.

Sentences containing dosage instructions were obtained from a private set of scanned typewritten drug prescriptions. After anonymization of the drug prescriptions, state-of-the-art Optical Character Recognition (OCR) software was used[19]. Low-quality sentences were then discarded from the output

[18] The dataset is available at https://huggingface.co/datasets/Posos/MedNERF.
[19] https://cloud.google.com/vision

of the OCR and the sentences containing dosage instructions were manually identified. For the purpose of this thesis only 100 sentences have been randomly sampled and made public through the MedNERF dataset, which is intended to be a test and not a training dataset.

The annotations of the medical sentences use the n2c2 labels `DRUG`, `STRENGTH`, `FREQUENCY`, `DURATION`, `DOSAGE` and `FORM`. The `ADE` (Adverse Drug Event) label was dropped since it is very rare that such entities are present in drug prescriptions. We also discarded the `ROUTE` and `REASON` labels following Frei, Frei-Stuber, and Kramer (2022) because of either their ambiguous definition or the lack of diversity of the matching samples. A total of 406 entities were annotated in 100 sentences (cf. Table 10.1).

NER Tag	Count
`DRUG`	67
`STRENGTH`	51
`FREQUENCY`	76
`DURATION`	43
`DOSAGE`	76
`FORM`	93
Total	406

Table 10.1 – Distribution of labels in MedNERF.

Another German dataset with similar annotation guidelines was released by Frei, Frei-Stuber, and Kramer (2022). It consists of 30 sentences from physicians annotated with the same guidelines as n2c2. Table 10.2 provides statistics about the different datasets used in this paper.

dataset	lang.	sent.	entities
n2c2	en	16,656	65,495
GERNERMED-test	de	30	119
MedNERF	fr	100	406

Table 10.2 – Statistics about the datasets.

Randomly sampled examples for each of those datasets are shown in Table 10.3. It must be noted that the French examples, drawn from drug prescriptions are significantly different from English and German examples, which are drawn from usually more verbose clinical reports. However, the fourth and fifth English examples show that n2c2 contains some examples that are similar to prescription instructions. The results will show that the different methods are able to generalize from English to French, despite the differences between the domains.

10.2 Controlled comparison of CTL and translation

This section focuses on a controlled comparison of cross-lingual transfer and translation-based methods. There are three methods to compare in total, as described in Section 2.3.3:

- Cross-lingual Transfer Learning (CTL): where a multilingual language model is trained on the original English dataset and evaluated on MedNERF
- `translate-train`: where a French language model is trained on a translated train set (with labels aligned with a word alignment tool) and evaluated on MedNERF
- `translate-test`: where an English language model is trained on the original train set and evaluated on the translated MedNERF, with the twist that labels must be aligned after having made the prediction on the translated test set.

Examples from n2c2
The patient's agitation was managed with **nightly**$_{\text{Frequency}}$ **haldol**$_{\text{Drug}}$ with **as needed**$_{\text{Frequency}}$ **haldol**$_{\text{Drug}}$ as well.
Improvement in clinical status was noted overnight and his **morphine**$_{\text{Drug}}$ drip was discontinued.
- hold all **antihypertensives**$_{\text{Drug}}$; plan to add back slowly at reduced doses and varying schedule - rule out MI - **bolus**$_{\text{Dosage}}$ **NS**$_{\text{Drug}}$ to maintain MAP $>$ 60 with caution given ESRD and oliguric.
folic acid$_{\text{Drug}}$ **1 mg**$_{\text{Strength}}$ **Tablet**$_{\text{Form}}$ Sig: **One (1)**$_{\text{Dosage}}$ **Tablet**$_{\text{Form}}$ PO **DAILY (Daily)**$_{\text{Frequency}}$.
Iron$_{\text{Drug}}$ **50 mg**$_{\text{Strength}}$ **Tablet**$_{\text{Form}}$ Sustained Release Sig: **One (1)**$_{\text{Dosage}}$ **Tablet Sustained Release**$_{\text{Form}}$ PO **once a day**$_{\text{Frequency}}$.
Examples from GERNERMED test set
Das **Eplerenon**$_{\text{Drug}}$ ist wegen Ihrer Herzinsuffizienz. Da können wir jetzt auf **50 mg**$_{\text{Strength}}$ p.o. 1-0-0 augmentieren. *Eplerenon is for your heart failure. We can now augment to 50mg p.o. 1-0-0.*
Zur Optimierung der Herzinsuffizienztherapie wurde die Dosis von **Sacubitril / Valsartan**$_{\text{Drug}}$ auf **97 / 103 mg**$_{\text{Strength}}$ in **Tablettenform**$_{\text{Form}}$ mit Einnahme **am Morgen und am Abend**$_{\text{Frequency}}$ erweitert. *To optimize heart failure therapy, the dose of sacubitril / valsartan was extended to 97 / 103 mg in tablet form with intake in the morning and evening.*
Bei bekannter koronarer Herzerkrankung sollte **lebenslang**$_{\text{Duration}}$ **Acetylsalicylsäure**$_{\text{Drug}}$ **100 mg**$_{\text{Strength}}$ **morgens täglich**$_{\text{Frequency}}$ in oraler Applikation eingenommen werden. *In cases of known coronary artery disease, acetylsalicylic acid 100mg should be taken orally daily in the morning as a lifelong treatment.*
Examples from MedNERF
TRAMADOL / PARACETAMOL$_{\text{Drug}}$ **37,5mg / 325mg**$_{\text{Strength}}$ *TRAMADO/PARACETAMOL 37,5mg/325mg*
AMLODIPINE$_{\text{Drug}}$ **5 mg**$_{\text{Strength}}$; **cpr**$_{\text{Form}}$ **1**$_{\text{Dosage}}$ **comprimé**$_{\text{Form}}$ **matin**$_{\text{Frequency}}$ **1**$_{\text{Dosage}}$ **comprimé**$_{\text{Form}}$ **soir**$_{\text{Frequency}}$ *AMLODIPINE 5mg; tab 1 tablet in the mording 1 tablet in the evening*
DOLIPRANETABS$_{\text{Drug}}$ **1000 MG**$_{\text{Strength}}$ **CPR PELL**$_{\text{Form}}$ PLQ / 8 (**Paracétamol**$_{\text{Drug}}$ **1.000 mg**$_{\text{Strength}}$ **comprimé**$_{\text{Form}}$) *DOLIPRANETABS 1000mg TAB PLQ / 8 (Paracetamol 1,000mg tablet)*
ACIDE ACETYLSALICYLIQUE$_{\text{Drug}}$ (**sel de lysine**$_{\text{Drug}}$) **75 mg**$_{\text{Strength}}$ **pdre p sol buv sach**$_{\text{Form}}$ (**KARDEGIC**$_{\text{Drug}}$) *ACETYLSALICYLIC ACID (lysine salts) 75mg oral powder for suspension (KARDEGIC)*
1$_{\text{Dosage}}$ **sachet**$_{\text{Form}}$ **matin midi et soir**$_{\text{Frequency}}$ si besoin *1 packet in the morning, at noon, and in the evening, if needed*

Table 10.3 – Examples from the different NER datasets used, with English translation in italics when needed.

While CTL only relies on one language model, translation-based methods require a translation model and a word alignment tool. Since MedNERF is designed for zero-shot CTL, the translation and alignment models cannot be chosen based on downstream performances. They can only be selected based on intrinsic evaluation. The following section describes this selection process.

10.2.1 Pre-selecting translation and alignment models

To select the translation and alignment models, this thesis relies on their inherent translation performances with BLEU (Papineni et al., 2002) and COMET (Rei et al., 2020) for translation models and alignment error rate (AER) for word alignment tools. However, the choice of the evaluation data raises a few issues. There are some parallel data in the biomedical, which is not strictly the clinical domain but close to it[20]. It allows us to try to fine-tune the translation models and word alignment tools to adapt them to the task at hand, and it also provides a way for evaluating intrinsic performances in-domain, but only for the translation models because no word alignment evaluation data is available in the clinical or biomedical domain.

10.2.1.1 Translation models

The automated translation of the n2c2 dataset from English to French and German is performed with the following transformer-based machine translation algorithms: Opus-MT (Tiedemann and Thottingal, 2020a) and FAIR (Ng et al., 2019) which can be fine-tuned on a corpus of bilingual medical texts proposed in the BioWMT19 challenge [21] (Bawden et al., 2019). The UFAL dataset is used, which is a collection of medical and general domain parallel corpora in 8 languages paired with English, as well as Medline, a dataset containing the titles and abstracts of scientific publications from Pubmed in English and a foreign language.

Since the UFAL dataset is orders of magnitude larger than Medline, it is downsampled to have equal proportions of sentences coming from Medline, from the medical part of UFAL and from UFAL general data. This results in approximately 90k sentences for the German translation models and 164k sentences for translation into French.

The Opus-MT model (Tiedemann and Thottingal, 2020b) is fine-tuned for translation to French and German and the FAIR model (Ng et al., 2020) only for translation to German as there is not any version of it is available in French.

The quality of the different translation models is measured on the Medline test set of the BioWMT 19 challenge and on the Khresmoi dataset (Dušek et al., 2017). Results are presented in Tables 10.4 and 10.5.

model	BioWMT19		Khresmoi	
	BLEU	COMET	BLEU	COMET
FAIR	<u>32.8</u>	0.628	**33.7**	**0.667**
+ ft	**34.2**	**0.734**	<u>32.4</u>	<u>0.666</u>
Opus	32.2	0.651	<u>32.4</u>	0.608
+ ft	32.5	<u>0.700</u>	30.5	0.619

Table 10.4 – Evaluation of the translation models from English to German. Best model in bold and second underlined. ft for finetuned.

The FAIR fine-tuned model is chosen as the best translation model for English to German translation. The Opus-MT fine-tuned model is selected as the best translation model for English to French translation.

[20] This thesis differentiates the biomedical domain which contains scientific content about health issues, like articles drawn from PubMed, and the clinical domain, which is content that is written by a healthcare professional during their practice.

[21] The links of the datasets of the BioWMT19 challenge are available on its page https://www.statmt.org/wmt19/biomedical-translation-task.html

model	BioWMT19		Khresmoi	
	BLEU	COMET	BLEU	COMET
Opus	35.9	0.672	**48.0**	**0.791**
+ ft	**36.7**	**0.786**	46.5	**0.791**

Table 10.5 – Evaluation of the translation models from English to French.

The same pre-selection evaluation is performed for the `translate-test` approach. The `translate-test` method needs translation algorithms from German and French to English. Similarly to Section 10.2.1.1 the Opus-MT and FAIR algorithms are fine-tuned on the same medical datasets. The COMET and BLEU scores obtained are presented in Tables 10.6 and 10.7. These scores are used to select the best translation model for the `translate-test` approach and they lead to the same model choices as for the `translate-train` method.

model	BioWMT19		Khresmoi	
	BLEU	COMET	BLEU	COMET
FAIR	<u>38.2</u>	0.538	**47.1**	**0.764**
+ ft	**38.5**	**0.675**	<u>46.8</u>	**0.764**
Opus	35.3	0.587	43.6	0.723
+ ft	38.1	<u>0.640</u>	44.3	0.729

Table 10.6 – Evaluation of the translation models from German to English. Best model bold and second underlined. ft for finetuned.

model	BioWMT19		Khresmoi	
	BLEU	COMET	BLEU	COMET
Opus	33.9	0.721	**48.3**	0.798
+ ft	**36.3**	**0.749**	48.0	**0.799**

Table 10.7 – Evaluation of the translation models from French to English. Best model bold. ft for finetuned.

10.2.1.2 Word alignment tools

`fast_align` and `awesome-align` are two popular choices for word alignment (Yarmohammadi et al., 2021). Since `awesome-align` can be fine-tuned on parallel data, it is also evaluated with further fine-tuning similar to translation models.

Choosing the right alignment models for the task can be tricky. While parallel corpora for fine-tuning `awesome-align` might be available in several languages and domains, annotated word alignment on parallel data is more scarce. Annotated word alignment test data is not available in the clinical domain. The best alignment models can thus only be selected based on performance on a general-domain dataset. `awesome-align` pre-trained on general-domain data is preferred in French, and the same model with further fine-tuning on biomedical data is selected for German.

Table 10.9 summarizes the choices of the best translation and alignment methods.

10.2.2 Implementation details

All models were written in Pytorch using the Huggingface librairies (Wolf et al., 2020) and were fine-tuned using the AdamW optimizer (Loshchilov and Hutter, 2019) and a learning rate of $6 \cdot 10^{-6}$ with linear decay. We used 4 epochs for the translation models and 8 epochs for the NER models. The NER models were trained on a single GPU (Nvidia GeForce RTX 2070 with 8GB of RAM) for approximately one hour.

model	fr	de
FastAlign	10.5	27.0
AWESOME from scratch	5.6	17.4
+ ft on clinical	4.7	15.4
AWESOME pre-trained	**4.1**	15.2
+ ft on clinical	4.8	**15.0**

Table 10.8 – Average Error Rate (AER) for various aligners.

lang	translation	alignment
fr	Opus ft	AWESOME
de	FAIR ft	AWESOME pt+ft

Table 10.9 – Pre-selected translation and alignment models.

`fast_align` was applied asymmetrically, by mapping words from source language (English) to target language. Although it might increase alignment score, symmetrization was not used, because it might remove important links during the labels projection step.

`awesome-align` was used with softmax (instead of the alternative α-entmax function) and without the optional consistency optimization objective. For completeness we added the results with the consistency optimization objective (w/ co lines) in the tables of Appendix 4 and we observed that they did not improve the NER scores. The base model used is mBERT as in the original paper (Dou and Neubig, 2021). The pre-trained version of `awesome-align` used is the one provided by the authors, fine-tuned on general-domain parallel data. Throughout the paper, in tables, "AWESOME" designates this latter pre-trained version. "AWESOME ft" is `awesome-align` with the raw mBERT model fine-tuned on the clinical parallel data only and "AWESOME pt+ft" is the pre-trained model, fine-tuned again on the clinical parallel data.

10.2.3 Results

The previous section described the selection of the translation and alignment models beforehand for a fair comparison between translation-based methods and CTL. This section details the results of this latter comparison shown in Table 10.10 with F1-scores on the target sets for the different methods. The translation and alignment models providing the best test scores are also shown for comparison, revealing what can be missed with the pre-selection.

CTL with a sufficiently large mLM provides the best results. When compared with translation-based methods with pre-selected translation and alignment models, CTL with XLM-R models gives higher scores, except for XLM-R Base in German.

On the other hand, it seems that using an English NER on a translated version of the test set provides the best results with a small model like distilmBERT. DistilmBERT might be better as a monolingual model than as a multilingual one. In the same vein, XLM-R Base struggles in German, while its large version does not. Small language models underperform in CTL and their generalization ability is not sufficient compared to translation-based methods. A first takeaway is consequently that translation-based methods should be favored with small language models.

The `translate-test` method is consistently outperformed by `translate-train` for large models. Translation and alignment errors only harm the training set of `translate-train`, which does not prevent a large model from generalizing despite some errors in the training data, while errors of translation or alignment in `translate-test` are directly reflected in the test score. In the rest of this analysis, we will consequently compare CTL only with `translate-train`.

Providing a large enough mLM, CTL outperforms `translate-train` and `translate-test` with pre-selected translation and word alignment models. However, choosing the translation and the alignment model beforehand does not lead to the best results. Eventually, Table 10.10 also shows that, to the exception of XLM-R Base in French, there always exists a pair of translation and

method	fr	de
distilmBERT		
CTL	$65.9_{\pm 3.3}$	$64.6_{\pm 2.4}$
translate-train select.	$66.5_{\pm 1.9}$	$\mathbf{68.3}_{\pm 1.3}$
translate-test select.	$\mathbf{69.2}_{\pm 1.4}$	$\mathbf{68.3}_{\pm 1.8}$
translate-train best	$\underline{69.2}_{\pm 1.2}$	$\underline{69.2}_{\pm 1.2}$
translate-test best	$\underline{69.7}_{\pm 1.5}$	$\underline{68.3}_{\pm 1.8}$
XLM-R Base		
CTL	$\mathbf{79.1}_{\pm 0.8}$	$72.2_{\pm 0.7}$
translate-train select.	$74.6_{\pm 0.9}$	$\mathbf{73.7}_{\pm 0.9}$
translate-test select.	$74.2_{\pm 1.6}$	$72.7_{\pm 0.8}$
translate-train best	$78.6_{\pm 0.5}$	$\underline{74.8}_{\pm 1.0}$
translate-test best	$74.4_{\pm 1.3}$	$72.7_{\pm 0.8}$
XLM-R Large		
CTL	$\mathbf{77.9}_{\pm 1.7}$	$\mathbf{78.5}_{\pm 0.4}$
translate-train select.	$76.5_{\pm 0.7}$	$77.4_{\pm 1.3}$
translate-test select.	$75.3_{\pm 0.9}$	$76.1_{\pm 2.8}$
translate-train best	$\underline{78.0}_{\pm 0.5}$	$\underline{79.4}_{\pm 1.3}$
translate-test best	$75.3_{\pm 0.9}$	$76.1_{\pm 2.8}$

Table 10.10 – Comparing the three methods with pre-selected translation and alignment models (select.). Best-performing pairs are provided for comparison and are underlined when better than CTL.

alignment models that leads to a better score for the `translate-train` method over CTL. This agrees with Yarmohammadi et al. (2021) and encourages practitioners to explore different methods to perform cross-lingual adaptation.

10.3 Influence of translation and alignment models

The choice of the translation and alignment models can have an important impact on the final NER performances as shown in Table 10.10. This section studies their impact in detail.

10.3.1 The impact of the choice of translation models

For the purpose of this section, a German NER model was trained using the Opus model instead of the FAIR model for translating the training set into German. Using a worse model (see Table 10.4) for translation leads to lower NER scores as shown in Table 10.11: the NER model based on the FAIR translation beats by more than 2 points the one using the Opus translation, whatever aligner is used.

While choosing between different translation models (like Opus or FAIR) based on their translation scores on in-domain data seems to provide the best results, deciding between the fine-tuned version of a translation model and the base one by comparing the BLEU or COMET scores on biomedical data does not guarantee the best downstream F1 score as Table 10.12 shows. The translation model was fine-tuned on biomedical data, which improved the intrinsic results on the BioWMT19 translation dataset. But this dataset belongs to a specific biomedical sub-domain (PubMed abstracts), and fine-tuning might not improve translation for the clinical sub-domain of the NER dataset.

The takeaway is that, while a small gain in translation accuracy (obtained with further fine-tuning) might not necessarily improve the result of the `translate-train` approach, a completely different model (like FAIR with respect to Opus) has more chance to improve cross-lingual adaptation.

aligner	Opus f1	FAIR f1
FastAlign	$70.9_{\pm 1.8}$	$\mathbf{72.8}_{\pm 1.6}$
AWESOME	$72.2_{\pm 1.7}$	$\mathbf{73.1}_{\pm 1.3}$
AWESOME ft	$71.1_{\pm 1.2}$	$\mathbf{74.1}_{\pm 1.1}$
AWESOME pt+ft	$71.2_{\pm 1.1}$	$\mathbf{74.1}_{\pm 1.3}$

Table 10.11 – `translate-train` in German with XLM-R Base using either fine-tuned or AWESOME model.

aligner	base	fine-tuned
FastAlign	$78.2_{\pm 0.8}$	$\mathbf{78.6}_{\pm 0.5}$
AWESOME	$\mathbf{76.4}_{\pm 2.0}$	$74.2_{\pm 0.9}$
AWESOME ft	$\mathbf{74.6}_{\pm 1.6}$	$74.5_{\pm 1.6}$
AWESOME pt+ft	$75.8_{\pm 1.7}$	$\mathbf{76.3}_{\pm 1.0}$

Table 10.12 – `translate-train` in French with XLM-R Base using either fine-tuned or base Opus model.

10.3.2 The impact of the choice of alignment models

While choosing a translation model based solely on intrinsic performance should not harm downstream cross-lingual adaptation performances, the choice of the alignment model seems more tricky. Based on intrinsic performances like the error rate on annotated alignment (Table 10.8), `awesome-align` seems to be the right aligner for the task. However, while it provides better downstream results than `fast_align` in German (Table 10.11), it does not hold for French (Table 10.12).

Table 10.13 shows that using different aligners leads to different levels of accuracy according to the types of entities we want to retrieve. While the global F1 score suggests that `fast_align` is better suited for the cross-lingual adaptation, looking at the detailed results for each entity type shows that the gap is mainly due to the `FREQUENCY` class on which `awesome-align` performs poorly. But this is not the case in other classes.

`FREQUENCY` entities are usually more verbose than drugs or dosages. Table 10.14 shows that `fast_align` makes obvious errors like aligning "240 mg" to "in morning and night", but `awesome-align` can miss the preposition when aligning "daily" with "jour" instead of "par jour", leading eventually to a consequent score drop.

The choice of the alignment model must thus be made more carefully than the translation one. Intrinsic performances of alignment models are not sufficient information. Some additional post-processing might be needed, as in Yarmohammadi et al. (2021), where `awesome-align` gives better results, but entities that are split by the aligner like "au moment du coucher" in Table 10.14 are merged by including all words in between. This would work in that particular case but could cause problems in others, particularly for languages where the word order is different.

aligner	Freq.	Strength	Drug	f1
FastAlign	**72.0**	89.7	**83.2**	**78.6**
AWESOME	50.2	**92.5**	82.2	74.2

Table 10.13 – Comparison of `fast_align` and `awesome-align` (pre-trained only) for three different entity types (F1-score), for `translate-train` with XLM-R Base on MedNERF with Opus fine-tuned.

original	FastAlign	AWESOME
in morning and night	le matin et la nuit et 240 mg	**le matin et la nuit**
daily	**par jour**	jour
once a day	**une fois par jour**	fois par jour
at bedtime	**au moment du coucher**	au // coucher

Table 10.14 – Examples of frequencies transformed with translation and alignment. Bold indicates the right annotation and // indicates that the entity has been split.

10.4 Using realignment

With the right translation and alignment model, it seems that CTL can be outperformed by the `translate-train` method. However the latter relies on additional resources: a translation and an alignment model, trained on parallel data. This parallel data could also be used to re-align the representations of the multilingual models used in CTL.

To improve a multilingual language model with parallel data, it is trained for a contrastive alignment objective following the realignment method from Wu and Dredze (2020) with strong alignment that was already described in Section 4.3.2.1. Words aligned with `awesome-align` are trained to have more similar representations than random in-batch pairs of words. After this realignment step, CTL can be applied.

model	fr	de
distilmBERT	$65.9_{\pm 3.3}$	$64.6_{\pm 2.4}$
+ realign	$\underline{66.4}_{\pm 1.4}$	$\underline{67.9}_{\pm 1.5}$
translate-train best	$\mathbf{69.2}_{\pm 1.2}$	$\mathbf{69.2}_{\pm 1.2}$
XLM-R Base	$\mathbf{79.1}_{\pm 0.8}$	$72.2_{\pm 0.7}$
+ realign	$76.7_{\pm 0.7}$	$\underline{\mathbf{75.8}}_{\pm 1.3}$
translate-train best	$78.6_{\pm 0.5}$	$74.8_{\pm 1.0}$
XLM-R Large	$77.9_{\pm 1.7}$	$78.5_{\pm 0.4}$
+ realign	$\underline{\mathbf{78.8}}_{\pm 1.6}$	$78.3_{\pm 1.6}$
translate-train best	$78.0_{\pm 0.5}$	$\mathbf{79.4}_{\pm 1.3}$

Table 10.15 – F1 scores for CTL from scratch and CTL with realignment. Best F1-score in bold. Results underlined show improvement of realignment over CTL.

Results in Table 10.15 show that while realignment does not systematically provide improvement over CTL as observed by Wu and Dredze (2020), it does significantly boost results in some cases, allowing to outperform the best `translate-train` baseline in German for XLM-R Base and in French for XLM-R Large. This, yet again, encourages practitioners to explore different methods, including realignment to perform cross-lingual adaptation.

10.5 Using domain-specific language models

This section answers what might be the most important consideration of this chapter: In a given domain, can domain-specific language models help translation-based methods outperform CTL?

This section evaluates the relevance of using language-specific models like CamemBERT (Martin et al., 2020) or GottBERT (Scheible et al., 2020) on the translated version of the training dataset or language and, more importantly, domain-specific models like DrBERT (Labrak et al., 2023) or medBERT.de (Bressem et al., 2023) which are BERT models fine-tuned on medical corpora in, respectively, French and German. Table 10.16 and 10.17 contains the results of the `translate-train`

method for the best translation/alignment algorithms pair and for the pre-selected one, compared to using XLM-R Base.

model	pre-selected	best
CamemBERT Base	$73.5_{\pm 1.5}$	$76.7_{\pm 0.9}$
DrBERT 7GB	$70.7_{\pm 1.3}$	$73.5_{\pm 1.4}$
DrBERT Pubmed	$\mathbf{76.1}_{\pm 1.3}$	$\mathbf{78.8}_{\pm 1.4}$
XLM-R Base	$74.6_{\pm 1.9}$	$78.6_{\pm 0.5}$

Table 10.16 – Comparison of domain and language-specific models for translate-train in French.

model	pre-selected	best
GottBERT	$\mathbf{75.5}_{\pm 1.4}$	$\mathbf{76.6}_{\pm 0.8}$
medBERT	$72.7_{\pm 0.5}$	$75.0_{\pm 1.6}$
XLM-R Base	$73.7_{\pm 0.9}$	$74.8_{\pm 1.0}$

Table 10.17 – Comparison of domain and language-specific models for translate-train in German.

The `translate-train` approach allows us to rely on models that are specific to the language and domain of the target evaluation. However, Table 10.16 and 10.17 show that their use does not always bring significant improvement over XLM-R Base. The performances of these models can be explained by the quantity of training data used. As shown in Table 10.18 XLM-R models are indeed trained on 2.5 TB data while DrBERT and medBERT.de use less than 10GB data, which can explain their low score. Besides, the language-specific models CamemBERT and GottBERT are trained with more data (138 GB and 145 GB) and achieve better performances, even beating XLM-R in German.

model	params (M)	emb. (M)	train (GB)
Multilingual models			
distilmBERT	135	92	42
XLM-R Base	278	192	2.5k
XLM-R Large	560	256	2.5k
Language-specific models			
CamemBERT (fr)	111	25	138
GottBERT (de)	126	40	145
Clinical models			
medBERT (de)	109	23	10
DrBERT 7GB (fr)	111	25	7.4
DrBERT PubMed (fr)	109	23	28

Table 10.18 – Size of the different base models. Although distilmBERT has more parameters than CamemBERT, it must be noted that it has also more words in its vocabulary, due to its multilingual nature. Hence most of its parameters are embedding weights that are not necessarily used in our experiments as they might be embeddings of words from other languages. So in our setting, distilmBERT can be considered a smaller model than CamemBERT and GottBERT despite the higher number of parameters.

Finally, it must be noted that the best `translate-train` model in French, DrBERT Pubmed, is actually pre-trained on the English PubMed dataset and then on French clinical texts, which suggests that multilingual models should be preferred, even with a translation-based cross-lingual adaptation. If multilingual models provide better results, even in translation-based methods, this thesis strongly advocates for the creation of multilingual domain-specific models. Those could improve translation-based methods of CTL in specific domains.

10.6 Conclusion

This chapter shows that cross-lingual transfer with multilingual LMs is efficient for a domain-specific task like clinical NER, giving comparable results with translating the training set. But CTL has the advantage of working off-the-shelf, while translation-based methods require choosing the translation and alignment models carefully. Selecting these models based on intrinsic domain-specific values, like fine-tuning scores on clinical parallel data, or using a domain-specific language model does not provide significantly better downstream results in the target language. The selection of the alignment model was shown to be particularly crucial, and the results of `translate-train` could probably be improved by post-processing the alignment. CTL also has a margin of progression as realigning the representations of MLLMs can increase the results dramatically in some cases.

It is also worth noting that training on translated data provides better results than translating at inference time. The `translate-test` approach should then be used only when large multilingual models cannot be used. While training on translated data allows to leverage domain-specific monolingual language models, those latter models can give better results over multilingual models like XLM-R only if pre-trained with sufficient data.

Pre-training a multilingual LMs with only clinical data is a good lead for further improvements in clinical cross-lingual transfer. While the results show that using a domain-specific monolingual model in `translate-train` or `translate-test` is not on par with general-purpose multilingual models, they also show that the French clinical model DrBERT provides the best results for `translate-train` when it uses the English biomedical model PubmedBERT as initialization.

This thesis finally advocates for the release of more non-English clinical datasets annotated with similar guidelines as English (or other) ones. Even a relatively small dataset like MedNERF or the GERNERMED test set is crucial to evaluate cross-lingual adaptation in the clinical domain.

Chapter 11

Conclusion

The first contribution of this thesis is to provide a better understanding of multilingual embeddings, by comparing how methods used to build static and contextualized multilingual embeddings differ. The second contribution is to evaluate how these multilingual representations can be applied to the clinical domain in a zero-shot cross-lingual setting.

This conclusion summarizes this thesis before delving into the broader lessons learned through these contributions and what they entail for further research.

11.1 A better understanding of multilingual embeddings

This thesis links static and contextualized multilingual embeddings, finding that the multilingual alignment targeted by static embeddings is strongly correlated with the cross-lingual transfer abilities of contextualized embeddings. While static embedding can also be trained in a language-agnostic manner like mBERT or XLM-R, those latter contextualized models provide better multilingual alignment than static embeddings that were explicitly trained to be aligned. Thus, this thesis answers the following open questions, among others, raised in background chapters:

Open Question 2.3. *How do contextualized and static multilingual embeddings compare in terms of word-level multilingual alignment?*

Open Question 4.2. *Is multilingual alignment correlated with cross-lingual transfer?*

Open Question 4.1. *Does language-agnostic training only lead to cross-lingual abilities for Transformer encoder-only models? What would happen if we performed the same kind of training for learning static word embeddings or generative LMs?*

11.1.1 Contextualized embeddings should often be preferred

Static multilingual embeddings, particularly unsupervised ones, suffer from many limitations. This thesis shows that those limitations are not limited to any specific mapping-based method, and all tested mapping-based methods that rely on isometry fail to some degree in the following settings: for distant and low-resource languages, and between different domains.

On the other hand, contextualized models work quite well off the shelf. Although the literature shows that some low-resource languages can have degraded performance with cross-lingual transfer, this thesis suggests that re-alignment can help in that specific case, provided that the word pairs used for realignment are extracted accurately, with a bilingual dictionary rather than with a word alignment tool.

11.1.2 The link between alignment and CTL is established

Finding a link between multilingual alignment and CTL is not completely surprising. However, this thesis argues that this link still needed to be verified with extensive experiments, especially since

this link was questioned by the failure of realignment methods (Wu and Dredze, 2020) and the absence of consensus in the literature about the presence of multilingual alignment in models like mBERT despite their CTL abilities (Doddapaneni et al., 2021).

The results of Chapter 9 establish a clear link between cross-lingual transfer learning and multilingual alignment, and more precisely with strong alignment.

11.1.3 Some improvements proposed

While this thesis mainly provides a better understanding of multilingual embeddings, its contribution is not only analytical. It also proposes two new methods for obtaining multilingual embeddings.

Firstly, Chapter 6 demonstrates the importance of code-switching as a cross-lingual training signal. Appendix 3 expands on this result and proposes `CoSwitchMap` and `CoSwitchMap v2` as proof-of-concept that code-switching can be used to build competitive multilingual static embeddings.

Secondly, Chapter 8 proposes to extract contextualized pairs of translated words using a bilingual dictionary rather than probabilistic tools. Chapter 9 shows that using such pairs as training samples of a realignment method provides significant improvements over the baseline.

11.2 Applications to the clinical domain

The bulk of the contribution focuses on the general domain because annotated multilingual data is mainly available in such domain. However, this thesis ultimately aims at applying multilingual embeddings to the clinical domain. Having proposed a new dataset, MedNERF, for evaluating cross-lingual transfer in the clinical domain, it can thus answer the following question, at least for this specific dataset:

Open Question 5.1. *How do translation-based methods and cross-lingual transfer compare in the clinical domain? Is it possible to leverage domain-specific language models with translation-based methods?*

11.2.1 Static embeddings suffer from additional limitations in a cross-domain setting

While Chapter 7 verifies that static embedding have many limitations, it also shows that embeddings of different domains are harder to align for a specific reason: the specificities of their vocabulary distribution might require a partial alignment of the representations.

11.2.2 CTL with contextualized embeddings works well off-the-shelf

Contrary to static embeddings, multilingual Transformers like mBERT can still perform well when applied outside of their domain of pre-training. Chapter 10 shows that such models provide good results on a clinical NER task, competitive with translation-based baselines.

This thesis validates that multilingual contextualized embeddings can be used in a clinical setting, particularly in a zero-shot cross-lingual scenario, where the selection of a translation model might become tricky.

11.2.3 MedNERF: a new dataset for clinical cross-lingual evaluation

Besides carefully evaluating existing methods in the clinical domain, this thesis also produces MedNERF, a French dataset for evaluating cross-lingual transfer learning on a clinical NER task. This dataset is available at the following link as a HuggingFace dataset:

https://huggingface.co/datasets/Posos/MedNERF

The best-performing model (XLM-R Base) fine-tuned on English examples found in Chapter 10 is also released for anyone to use:

https://huggingface.co/Posos/ClinicalNER

11.3 Lessons learned and further research

While this thesis answers some specific research questions, it also provides some lessons about the study of multilingual representations and language models in general, as well as leads for future research.

11.3.1 Studying the dynamic multilingual models

The CTL abilities of models like mBERT are surprising. But trying to explain this is difficult as the literature has found many factors that may influence CTL (cf. Section 4.2). Trying to test those factors in a rigorous setting is costly. The ideal approach would be to perform several pre-training runs with different characteristics, e.g. with and without allowing vocabulary overlap. But pre-training is costly. One solution is to perform the experiments on smaller models, as Chapter 6 does with code-switching or like similar works do for other factors (Dufter and Schütze, 2020). But there is no guarantee that smaller settings generalize well to larger ones.

This thesis argues that one should first provide a better understanding of existing representations built by models like mBERT before trying to guess which factor to test in a controlled experiment on pre-training. Studying the geometry of those multilingual representations might then provide sufficient insights on which factors to control for in pre-training. The results of Chapter 6 typically suggest that testing for artifacts of the pre-training data like code-switching might be a good lead, despite some disappointing early negative results in Chapter 6 on a small scale. Chapter 9 also shows that multilingual alignment is stronger in deeper layers, suggesting that the multilinguality of mBERT and others does not necessarily come from lexical overlap but maybe from higher-level features shared between languages.

But this thesis also advocates for doing more than just probing frozen pre-trained model. Indeed, Chapter 9 shows that fine-tuning multilingual Transformers on monolingual data can have unsuspected effects on its multilingual properties. This advocates for more research into the dynamic of fine-tuning multilingual models, instead of just probing frozen representations.

11.3.2 Scaling is not all you need

When it comes to CTL, there is an undeniable trend that shows that larger model (with more pre-training data and more parameters) often provide better results. This is verified throughout this thesis (Chapters 9 and 10) and in the literature (Ahuja et al., 2023; Philippy, Guo, and Haddadan, 2023). However, this thesis also shows that, at least for some tasks and languages, scaling is not the only way to improve downstream results. Chapter 9 shows that realignment works particularly well in some cases for small models like distilBERT, to the point that distilmBERT can match the performance of XLM-R Large with realignment on POS-tagging in Arabic with a gain of more than 15 points.

11.3.3 Towards building multilingual clinical models

Chapter 10 shows that general-domain multilingual models work better than clinical-domain monolingual models when used with a translation-based method for cross-lingual generalization. However, results can be improved, with a "roughly bilingual" clinical-domain model like DrBERT+Pubmed, which is a French clinical model initialized with the weights of an English biomedical model. This suggests that pre-training multilingual models from scratch or from already pre-trained models on multilingual and clinical data could bring better downstream performances in a clinical cross-lingual setting.

This thesis advocates for further research into simultaneously pre-training for specific domains and multilinguality.

11.3.4 Extending the analysis to generative models

Generative models were put aside in this thesis, due to a willingness to focus on embeddings but also because some other works show that generative models might not not be there yet in term of cross-lingual transfer (Ahuja et al., 2023).

However, evidence suggests that some of the previous conclusions might generalize well to generative models. Chapter 8 still includes an encoder-decoder model in its analysis: mBART. It shows that its encoder has aligned representations similarly to other encoder-only models. Furthermore, Li and Murray (2023) show that fine-tuning mT5 on XNLI increases the alignment of its inner representations, similarly to what we find with encoder-only models in Chapter 9.

This advocates for extending the contribution to generative models as future research.

Appendix

Chapter 1

Approximation of the Gromov-Hausdorff distance

The Gromov-Hausdorff distance provides an exact measure of the deviation from isometry between two metric sets. However, it is intractable to compute in practice. Thus this thesis, following previous work (Patra et al., 2019), uses an approximation (Chazal et al., 2009).

1.1 The Gromov-Hausdorff distance

To measure the deviation from isometry of two metric spaces, we must first be able to evaluate to what extent two aligned sets coincide. For that one can rely on the Hausdorff distance.

Definition 1.1. *Let $\mathcal{X}$ and $\mathcal{Y}$ be two compact subsets of a metric space $(\mathcal{Z}, d_Z)$. The Hausdorff distance is defined by:*

$$d_H^{\mathcal{Z}}(\mathcal{X}, \mathcal{Y}) = \max\left(\sup_{x\in\mathcal{X}} \inf_{y\in\mathcal{Y}} d_Z(x,y), \sup_{y\in\mathcal{Y}} \inf_{x\in\mathcal{X}} d_Z(x,y)\right) \tag{1.1}$$

The Hausdorff distance is the maximum distance between pairs of nearest neighbors. From that the Gromov-Hausdorff distance can be computed, and gives a theoretical measure of the deviation from isometry:

Definition 1.2. *Let $(\mathcal{X}, d_X)$ and $(\mathcal{Y}, d_Y)$ be two metric spaces. The Gromov-Hausdorff distance is defined by:*

$$d_{GH}((\mathcal{X}, d_X), (\mathcal{Y}, d_Y)) = \inf_{\mathcal{Z}, f, g} d_H^{\mathcal{Z}}(f(\mathcal{X}), g(\mathcal{Y})) \tag{1.2}$$

With $f : \mathcal{X} \to \mathcal{Z}$ and $g : \mathcal{Y} \to \mathcal{Z}$ isometries matching both metric spaces to a single metric space $(\mathcal{Z}, d_Z)$.

1.2 Approximation of the Gromov-Hausdorff distance

The Gromov-Hausdorff distance is the minimum over all isometric transformations of the Hausdorff distance. It measures how well two metric spaces can be aligned without deforming them. This distance, intractable to compute in practice, needs to be approximated. An approximation can be given by the Bottleneck distance between the persistence diagrams of the Rips complex filtrations of each metric space and it was shown to be a tight lower-bound for the Gromov-Hausdorff distance (Chazal et al., 2009), and was found to better correlate with the ability to align embedding with an orthogonal mapping than previously mentioned metrics (Patra et al., 2019).

1.2.1 Persistence diagram

For a more formal definition of persistence diagram, Rips complex and filtrations, the reader might refer to relevant literature (Chazal et al., 2009). A parameter t that varies from 0 to $+\infty$, allows to compute a graph for each embedding such that two points of an embedding that are at a distance smaller than $2t$ are linked by an edge. This graph is a Rips complex, or rather a simplified version of it because we only look at connected components, hence only edges, not higher-dimensional simplexes. We start with as many connected components as elements in the embedding ($t = 0$) and gradually decrease their number by merging them. This sequence of Rips complexes is called a filtration, on which we can compute a persistence diagram for each embedding: a list of points $(t_{\text{birth}}, t_{\text{death}})$ for each connected component that appears during the filtration recording the $t{=}t_{\text{birth}}$ at which it appears and the $t{=}t_{\text{death}}$ at which it is merged with another.

1.2.2 Bottleneck distance

Comparing the persistence diagrams for two embeddings allows us to measure to what extent they differ topologically. This is the Bottleneck distance.

Definition 1.3. *Given two multi-sets A and B in $\overline{\mathbb{R}}^2$, the Bottleneck distance is defined by:*

$$d_B^{\infty}(A, B) = \min_{\gamma} \max_{p \in A} ||p - \gamma(p)||_{\infty} \tag{1.3}$$

With γ ranging over bijections between A and B.

The Bottleneck distance between Rips filtration of metric spaces gives us a lower bound for the Gromov-Hausdorff distance.

Theorem 1.1. *From Chazal et al. (2009). For any finite metric spaces $(\mathcal{X}, d_X)$ and $(\mathcal{Y}, d_Y)$ and for any $k \in \mathbb{N}$:*

$$d_B^{\infty}(\mathcal{D}_k\mathcal{R}(\mathcal{X}, d_X), \mathcal{D}_k\mathcal{R}(\mathcal{Y}, d_Y)) \leq d_{GH}((\mathcal{X}, d_X), (\mathcal{Y}, d_Y)) \tag{1.4}$$

For the sake of simplicity, in this thesis, when is mentionned the "Bottleneck distance" or the "approximate GH distance" between two sets of embeddings, it will be actually referring to the Bottleneck distance between the persistence diagrams of the filtrations of Rips complexes built over those embeddings.

Chapter 2

Further experiments on the correlation between cross-lingual transfer and multilingual alignment

Section 9.1.4 compares correlation for different tasks, before and after fine-tuning, for English-target and target-English alignment and for the last and penultimate layer. These correlations where computed across several models, languages and seeds. From this correlation statistics, we have drawn three conclusions:

1. Strong alignment is better correlated with cross-lingual transfer than weak alignment.
2. The NLI task, because of its sentence-level nature, have a cross-lingual transfer that correlates better with the penultimate layer than the last one.
3. The results do not significantly attribute higher correlation of cross-lingual transfer with alignment before or after fine-tuning neither with English-target compared to target-English alignment.

This appendix chapter verifies that these conclusions hold when looking at the confidence intervals (Tab. 2.1 and Table 2.2). Confidence intervals are obtained using the Bias-Corrected and Accelerated (BCA) bootstrap method, where several subsets (2000) subsets of our 100 points for each measure of the correlation coefficient are sampled to obtain an empirical distribution of the correlation from which the confidence interval can be deduced (Efron and Tibshirani, 1994). Since this work is dealing with ordinal data (the rank in Spearman's rank correlation), bootstrap confidence intervals are expected to have better properties than methods based on assumptions about the distribution (Bishara and Hittner, 2017; Ruscio, 2008).

Is strong alignment significantly better correlated with cross-lingual transfer than weak alignment? comparing both tables cell-by-cell reveals that confidence intervals for the last layer before fine-tuning hardly never overlap, and when they do it is with a small overlap. So in the case of alignment of the last layer before fine-tuning, strong alignment is significantly better correlated with cross-lingual transfer than weak alignment. For other situations, confidence interval overlap. But the fact that strong alignment has almost systematically a higher correlation makes our correlation still relevant.

Does the penultimate layer correlate better than the last one for NLI? For this task, we observe that the confidence intervals of the penultimate and last layer do not overlap when the alignment is measured after fine-tuning. Otherwise, before fine-tuning, we can still observe that the measured correlation for the penultimate layer is systematically above the confidence interval for the last layer, except for target-English strong alignment.

Confidence intervals overlap significantly when comparing before and after fine-tuning, except in two cases. When looking at POS-tagging for the last layer, weak alignment after fine-tuning

task	layer	en-X		X-en	
		before	after	before	after
POS	last	0.58 (0.43 - 0.70)	0.84 (0.77 - 0.89)	0.63 (0.48 - 0.74)	0.83 (0.74 - 0.89)
	penult.	0.78 (0.68 - 0.85)	0.84 (0.76 - 0.89)	0.80 (0.71 - 0.87)	0.85 (0.79 - 0.90)
NER	last	0.69 (0.55 - 0.79)	0.72 (0.59 - 0.81)	0.75 (0.64 - 0.83)	0.84 (0.73 - 0.89)
	penult.	0.82 (0.73 - 0.88)	0.71 (0.58 - 0.81)	0.88 (0.83 - 0.92)	0.72 (0.58 - 0.82)
NLI	last	0.51 (0.32 - 0.67)	0.75 (0.61 - 0.85)	0.54 (0.36 - 0.68)	0.73 (0.59 - 0.83)
	penult.	0.74 (0.59 - 0.84)	0.95 (0.90 - 0.97)	0.79 (0.66 - 0.87)	0.94 (0.90 - 0.97)

Table 2.1 – 95% confidence interval for Spearman rank correlation between weak alignment and CTL, obtained with BCA bootstraping with 2000 resamples.

task	layer	en-X		X-en	
		before	after	before	after
POS	last	0.80 (0.73 - 0.86)	0.85 (0.81 - 0.88)	0.83 (0.77 - 0.87)	0.87 (0.83 - 0.91)
	penult.	0.86 (0.79 - 0.89)	0.85 (0.79 - 0.89)	0.87 (0.82 - 0.91)	0.86 (0.82 - 0.90)
NER	last	0.85 (0.78 - 0.90)	0.66 (0.53 - 0.77)	0.86 (0.82 - 0.90)	0.74 (0.65 - 0.82)
	penult.	0.87 (0.83 - 0.91)	0.75 (0.65 - 0.84)	0.88 (0.84 - 0.92)	0.76 (0.66 - 0.84)
NLI	last	0.74 (0.63 - 0.82)	0.81 (0.72 - 0.87)	0.89 (0.83 - 0.93)	0.84 (0.77 - 0.90)
	penult.	0.84 (0.79 - 0.88)	0.92 (0.87 - 0.95)	0.90 (0.86 - 0.93)	0.94 (0.90 - 0.96)

Table 2.2 – 95% confidence interval for Spearman rank correlation between strong alignment and cross-lingual transfer, obtained with BCA bootstraping with 2000 resamples.

gives a significantly better correlation than before, but this does not translate to strong alignment which correlates better with cross-lingual transfer overall. The same observation can be made about NLI for the penultimate layer. On the other hand, for the NER task, strong alignment after-fine tuning gives a significantly worse correlation than before. It is thus difficult to conclude on whether alignment before or after fine-tuning is better correlated to cross-lingual transfer.

Finally, comparing target-English and English-target alignment does not give significant results. If all other parameters are kept identical, every situation leads to an overlap between confidence intervals except for the last layer before fine-tuning for NLI, which might just be fortuitous since it's the second before last layer that correlates better with cross-lingual transfer for this task.

Chapter 3

Using code-switching for initializing mapping-based methods

3.1 Using code-switching for aligning different scripts

While Vecmap seems to be the unsupervised mapping-based method that provides the best and most stable results across various pairs of languages, it still largely fails on some specific language pairs. Since its self-learning procedure still allows to match supervised methods when using a supervised or weakly supervised initialization (Vecmap w/ identical in previous results), this section investigates whether better unsupervised initializations are possible for Vecmap.

This section proposes `CoSwitchMap` to extract code-switching situations in existing pre-training data and use them as a cross-lingual signal for learning a bilingual mapping. This reminds of pseudo-mixing methods evoked in Chapter 3, which randomly replace words by their translation in the training data of a word embedding in order to learn multilingual representations. But instead of artificially inducing code-switching in a supervised manner using a bilingual dictionary, this section proposes an unsupervised method to leverage naturally occurring code-switching situations. This answers the following open question raised earlier:

Open Question 3.1. *While there are unsupervised mapping-based approaches, Is it possible to design pseudo-mixing or explicitly joint approaches in an unsupervised setting?*

However, supervised pseudo-mixing approaches allow to learn an entire embedding matrix from artificially-induced code-switching situations, while the previous chapter has shown that natural code-switching might be too scarce of a phenomenon (cf. Section 6.3) to learn a brand new representation for each word in the entire vocabularies of two languages. Indeed, training a static embedding directly on monolingual data and hoping to pick up on the rare cross-lingual signal brought by code-switching does not work, as the failure of language-agnostic static embeddings proposed in previous chapters shows.

Therefore, this section proposes to modify the monolingual objective of a typicall word embedding to learn an orthogonal mapping. Thus, in term of training data, the proposed method can be seen as an unsupervised pseudo-mixing approach. But in term of what is effectively learnt, it is a mapping-based approach.

3.1.1 Identifying code-switching with different scripts

To identify code-switching situations we must find paragraphs that contain words coming from different languages. However, determining whether a word belongs to the vocabulary of one given language is not straightforward. Without resorting to additional resources like a dictionary, the vocabulary of one language can be obtained based on occurrences in a monolingual corpus. However,

if this monolingual corpus potentially contains code-switching, the vocabulary we would obtain might not help identify code-switching situations as it might include words from other languages.

If two languages are written using different scripts, most code-switching situations can be extracted by identifying paragraphs where the two scripts occur, using regular expressions with relevant character ranges. This method has, by design, a high recall, as it should only miss some situations where the word from one language is transcribed into the script of the other, which can still be seen as code-switching, or rather script-switching, situations. However, it can lack precision in some cases, because the same script can be used in different languages. For example, when extracting pairs of code-switched words involving English in a Chinese corpus, we might also retrieve German-Chinese pairs.

3.1.2 Learning a mapping with code-switching pairs and skip-gram

This thesis refers to a "code-switching pair" as a pair of two words from two different languages that were found in the same context. The goal is to leverage these pairs as a multilingual signal to learn a mapping matrix W that projects the words of a source language (src) to the target language (tgt). It must be noted that multi-word expressions are not getting a particular treatment, like in most word embedding algorithms. Code-switching pairs are pairs of words from different scripts found in the same sliding window of context. A multi-word expression like "Eurovision Song Contest" is broken down and each word that composes it will appear individually in pairs with Chinese neighboring words.

Given two monolingual embeddings for source and target languages obtained with skip-gram (Mikolov et al., 2013) or a variant like FastText (Bojanowski et al., 2017), two embedding matrices can be retrieved for each language: the central embedding of each word x_i, i.e. the embedding that is usually used in downstream application, and the context embedding $\tilde{x}_j$, used to embed context words in the skip-gram algorithm. The goal is to continue the training of skip-gram with code-switched words in order to learn a matrix W mapping the source embedding x_i^{src} to the target embedding x_j^{tgt}. During the training, the W matrix will be either applied to the context word or central word depending on the training pair. Thus, the embedding matrices are freezed and W is initialized with the identity matrix before being trained.

The original monolingual skip-gram loss from Mikolov et al. (2013) is the following :

$$L = -\frac{1}{|C|} \sum_{w_i \in C} \sum_{w_j \in \mathcal{N}(w_i)} \log P(w_j|w_i) \tag{3.1}$$

Where C is the corpus, w_i is a central word from the corpus, and w_j is a word found in $\mathcal{N}(w_i)$, the context window of the central word. $P(w_j|w_i)$ is computed with negative sampling as :

$$\begin{aligned} \log P(w_j|w_i) &= \log \sigma(\tilde{x}_j^\top x_i) \\ &+ \sum_{w_k \sim P_V}^{n} \log \sigma(-\tilde{x}_k^\top x_i) \end{aligned} \tag{3.2}$$

x_i is the embedding of w_i and $\tilde{x}_j$ is the context embedding of w_j. n negative examples of context words w_k are sampled randomly from a distribution P over the vocabulary V. Minimizing L in Equation 3.1 is maximizing the similarity of x_i with $\tilde{x}_j$ with respect to the similarity of x_i with any other randomly sampled word.

`CoSwitchMap` learns the mapping matrix W with a similar negative sampling loss, but replaces the source word embedding, either central or context, by their projection with W. The initial embedding obtained with skip-gram applied to monolingual corpora is frozen and the modified skip-gram loss is only computed for pairs of code-switched words. For a code-switching pair $(w_i^{\text{src}}, w_j^{\text{tgt}})$, where the central word w_i^{src} is in the source language script, and w_j^{tgt} is a context word in the target language script, the goal is to project w_i^{src} to the target language. The probability $P(w_j^{\text{tgt}}|w_i^{\text{src}})$

method	en-ar	en-ru	en-zh
VecMap	$36.4_{\pm 1.8}$	$\mathbf{49.1}_{\pm 0.4}$	$0.0_{\pm 0.0}$
w/ MUSE init.	$39.5_{\pm 0.7}$	$47.5_{\pm 2.5}$	$6.5_{\pm 12.6}$
w/ WP init.	$31.9_{\pm 15.9}$	$44.8_{\pm 3.3}$	$0.0_{\pm 0.1}$
w/ identical init.	$\underline{39.8}_{\pm 0.3}$	$\underline{48.9}_{\pm 0.2}$	$36.8_{\pm 0.8}$
`CoSwitchMap`	$\mathbf{39.9}_{\pm 0.1}$	$\underline{49.0}_{\pm 0.3}$	$\mathbf{37.9}_{\pm 0.9}$
supervised	*43.0*	*52.7*	*43.3*

Table 3.1 – Comparison of `CoSwitchMap` with other unsupervised mapping-based methods. The score is the top-1 accuracy of a nearest-neighbor search with CSLS criterion for BLI. Results are averaged over 5 seeds and the standard deviation is provided (except for the deterministic supervised baseline). Bold indicates the best score for a given language pair and all scores that are within the standard deviation of the best one are underlined.

becomes:

$$\begin{aligned}\log P(w_j^{\text{tgt}}|w_i^{\text{src}}) &= \log \sigma(\tilde{x}_j^{\text{tgt}\top} W x_i^{\text{src}}) \\ &+ \sum_{w_k^{\text{tgt}} \sim \mathcal{U}_{V_{\text{tgt}}}}^{n} \log \sigma(-\tilde{x}_k^{\text{tgt}\top} W x_i^{\text{src}})\end{aligned} \tag{3.3}$$

For the reversed case, where a code-Switching pair $(w_i^{\text{tgt}}, w_j^{\text{src}})$ is given, the central word is in the target language, and the context word in the source language. The mapping matrix must then be applied to the context embedding:

$$\begin{aligned}\log P(w_j^{\text{src}}|w_i^{\text{tgt}}) &= \log \sigma(\tilde{x}_j^{\text{src}\top} W^\top x_i^{\text{tgt}}) \\ &+ \sum_{w_k^{\text{src}} \sim \mathcal{U}_{V_{\text{src}}}}^{n} \log \sigma(-\tilde{x}_k^{\text{src}\top} W^\top x_i^{\text{tgt}})\end{aligned} \tag{3.4}$$

By enforcing the orthogonality of W, applying it to the source context embedding is actually equivalent to applying its inverse to the source central embedding. Using an orthogonal matrix also allows to preserve the distance between words from the source language. Thus, during the training steps, the mapping matrix W is orthogonalized after each update of the loss of a training batch as it was done in Lample et al. (2018) (cf. Equation 7.1).

3.1.3 Using `CoSwitchMap` as an initialization method

`CoSwitchMap` introduces a new way to learn a seed bilingual dictionary from code-switching. This seed dictionary can then be used as initialization for a self-learning loop. `CoSwitchMap` reuses the self-learning algorithm of VecMap.

Table 3.1 shows how `CoSwitchMap` fares compared to the other aforementioned mapping-based methods in a Bilingual Lexicon Induction (BLI) task. For the three language pairs tested, fully unsupervised mapping-based methods (WP, MUSE, and VecMap) are outperformed or matched by `CoSwitchMap`. The gap is the most significant for the English-Chinese pair, where fully unsupervised methods largely fail, while initialization with identical words scores slightly behind `CoSwitchMap`. For the two other language pairs, the differences are less pronounced but `CoSwitchMap` is still among the best-performing ones.

It must also be noted that the initialization provided by the code-switching training signal is significantly better than any other except the identical initialization for Arabic and Russian, and the original initialization for Russian. But `CoSwitchMap` always at least matches, if not outperforms, the best unsupervised baseline.

Moreoever, two things must be noted about the identical initialization. First, it might indirectly rely on code-switching, since the most frequently code-switched words will be present in the

vocabulary of both languages[22]. Second, `CoSwitchMap` still outperforms this baseline for the English-Chinese pair, suggesting that explicitly relying on code-switching can sometimes provide more accurate alignment.

Table 3.1 also shows the results of a competitive supervised baseline, from the same framework as VecMap Artetxe, Labaka, and Agirre, 2018b trained on a bilingual dictionary of 5,000 different words with their translations, distinct from those used for evaluation, but from the same origin (Lample et al., 2018). Being unsupervised, `CoSwitchMap` is unsurprisingly outperformed by the supervised baseline, but falls short only by a few points, from 3.1 to 5.6. The supervised method has the unfair advantage of relying on a training bilingual dictionary, which is similar to the test dictionary used for evaluating BLI.

Besides providing a new lead for improving unsupervised mapping-based methods, `CoSwitchMap` demonstrates that code-switching can be harnessed to learn cross-lingual word representations with little supervision. But is it really a fully unsupervised method? It only requires a list of characters (or rather a unicode range) for each script, but this can be seen as some level of supervision. Moreover, the method is not directly applicable to languages with same scripts. The following section thus intends to generalize `CoSwitchMap` to a larger setting.

3.2 Generalizing the method to languages with same script

This section introduces `CoSwitchMap v2`, a variant of `CoSwitchMap` that features two modifications. First, its way to extract code-switching pairs relies on the frequency of words. While it might be less accurate in extracting all and only code-switching pairs, it is a more versatile approach that namely allows to deal with languages written in the same script. Second, it proposes a simpler way to learn the mapping, by simplifying the modified skipgram objective proposed above. This later modification is welcomed in this context, as the method will be tested on more language and thus needs a faster way to be trained.

3.2.1 Extracting code-switching pairs according to frequency

When introducing `CoSwitchMap`, this thesis argued that recovering code-switching pairs using the frequency of words in monolingual corpora is tricky. Indeed, a given word might be frequent in a monolingual corpus because it is frequently code-switched. However, the findings of the previous chapter on the frequency of code-switching show that a word can be found in a corpus from another language with a frequency that is very rarely above 10^{-4}. Therefore, `CoSwitchMap v2` consider words that have a frequency above 10^{-4} in one language and not the others to be specific to that language. Then, when two words specific to two different languages are found in a same context, they can be considered to be part of a code-switching pair.

The script-based approach of `CoSwitchMap` lacks in precision because a same script can be used for different languages. On the other hand, the proposed frequency-base approach might lack in recall, as it ignores code-switching situations that involve rarer words.

3.2.2 A simpler way to learn the mapping

Because this new way to extract code-switching pairs allows to deal with more language pairs, this thesis attempted to make the learning of the initial mapping more efficient to allow more extensive experiments.

To remind the reader, `CoSwitchMap` uses a modified skipgram, where embedding are frozen and only a mapping W is updated in a modified skip-gram loss with negative sampling that writes as

[22]Only the most frequently code-switched words because vocabularies are usually truncated before alignment typically to 200,000 words

follows:

$$L = \frac{1}{|C|} \sum_{w_i, w_j \in C} \Bigg(\log \sigma \left(\tilde{x}_j^{\text{tgt}\top} W x_i^{\text{src}} \right) + \log \sigma \left(\tilde{x}_i^{\text{src}\top} W^\top x_j^{\text{tgt}} \right) + \sum_{w_k^{\text{tgt}} \sim \mathcal{U}_{V_{\text{tgt}}}}^{n} \log \sigma(-\tilde{x}_k^{\text{tgt}\top} W x_i^{\text{src}}) + \sum_{w_k^{\text{tgt}} \sim \mathcal{U}_{V_{\text{src}}}}^{n} \log \sigma(-\tilde{x}_k^{\text{src}\top} W x_j^{\text{tgt}}) \Bigg) \tag{3.5}$$

It is a sum over all source-target pairs (w_i, w_j) of four terms. On the first line, there are two terms for enforcing alignment between contextualized and central embeddings of the words of the given pair. The first one is for when w_i is viewed as the central word, and the second one for w_j as the central word. Similarly, the second line contains two term for the negative pair where random negative samples for the context words are drawn compared with central words.

In the original skip-gram algorithm, the context and central embeddings are not frozen. Thus, having negative samples allows to avoid degenerate solutions where all embeddings are collapsed to a single vector. But since `CoSwitchMap` is only learning an orthogonal mapping between central and context embeddings from different languages, those negative samples might no longer be necessary. Indeed, the enforced orthogonality of the matrix W avoids any degenerate solution. Therefore, one first simplification would be to get rid of the negative sampling.

Equation 3.5 then becomes:

$$L_1 = \frac{1}{|C|} \sum_{w_i, w_j \in C} \left(\log \sigma \left(\tilde{x}_j^{\text{tgt}\top} W x_i^{\text{src}} \right) + \log \sigma \left(\tilde{x}_i^{\text{src}\top} W^\top x_j^{\text{tgt}} \right) \right) \tag{3.6}$$

The two remaining terms inside the sum are simply enforcing two pairs of representations to be the most similar possible: $(\tilde{x}_j^{\text{tgt}}, W x_i^{\text{src}})$ and $(\tilde{x}_i^{\text{src}}, W^\top x_j^{\text{tgt}})$. Although it is not exactly an equivalent optimization problem, another way to enforce the similarity between those pairs would be to replace the function $\log \sigma(\cdot)$ by a simple scalar product, as follows:

$$L_2 = \frac{1}{|C|} \sum_{w_i, w_j \in C} \tilde{x}_j^{\text{tgt}\top} W x_i^{\text{src}} + \tilde{x}_i^{\text{src}\top} W^\top x_j^{\text{tgt}} \tag{3.7}$$

This thesis insists that it does not affirm that L_1 and L_2 are mathematically equivalent. But they simply have the same practical aim, which is to encourage the alignment of several pairs of representations, although they probably do not have the same exact optimal solution. L_1 has only been rewritten into L_2 because the latter boils down to an optimization problem that has a closed-form solution with the Procrustes algorithm used in many mapping-based method for embedding alignment. Indeed, minimizing L_2 is equivalent to optimizing for the following problem:

$$W^* = \arg\min_W ||A - BW||_F^2 \tag{3.8}$$

A and B are matrices of size $2N \times d$, d being the dimension of the embeddings and N the number of code-switching pairs. And they write as:

$$A = \begin{pmatrix} \tilde{x}_1^{\text{tgt}\top} \\ \dots \\ \tilde{x}_N^{\text{tgt}\top} \\ x_1^{\text{tgt}\top} \\ \dots \\ x_N^{\text{tgt}\top} \end{pmatrix}, B = \begin{pmatrix} x_1^{\text{src}\top} \\ \dots \\ x_N^{\text{src}\top} \\ \tilde{x}_1^{\text{src}\top} \\ \dots \\ \tilde{x}_N^{\text{src}\top} \end{pmatrix} \tag{3.9}$$

The problem stated in 3.8 is the orthogonal Procrustes Problem and its solution is given by the singular value decomposition (SVD) of $M = B\top A$. The SVD of provides three matrices U, S and V with $M = USV^\top$. And the solution of 3.8, is simply $W^* = UV^\top$.

method	en-ar	en-ru	en-zh
`CoSwitchMap v2`	$39.5_{\pm 0.3}$	$\underline{48.7}_{\pm 0.2}$	$37.2_{\pm 0.3}$
w/o frequency-based extraction	$39.4_{\pm 0.3}$	$\mathbf{49.0}_{\pm 0.3}$	$\underline{37.9}_{\pm 0.7}$
w/o simplified mapping learning	$\mathbf{39.9}_{\pm 0.2}$	$\underline{48.7}_{\pm 0.5}$	$\mathbf{38.4}_{\pm 0.8}$
w/o both (`CoSwitchMap`)	$\mathbf{39.9}_{\pm 0.1}$	$\mathbf{49.0}_{\pm 0.3}$	$\underline{37.9}_{\pm 0.9}$

Table 3.2 – Ablation analysis of the modification brought by `CoSwitchMap v2` over `CoSwitchMap`.

It is to be noted that there is never the need to hold the entire matrices A and B in memory. The matrix M, which is $d \times d$, can be directly computed from the embeddings as follows:

$$M = \sum_{j=1}^{N} x_j^{\text{src}} \tilde{x}_j^{\text{tgt}\top} + \sum_{i=1}^{N} \tilde{x}_i^{\text{src}} x_i^{\text{tgt}\top} \tag{3.10}$$

Therefore, instead of computing W with stochastic gradient descent with several epochs, `CoSwitchMap v2` proposes to compute the matrix M in one pass over the code-switching pairs, and then compute its SVD $M = USV^\top$ before computing W as $UV^\top$.

3.2.3 Ablation analysis on `CoSwitchMap v2`

To verify that the changes brought with `CoSwitchMap v2` are not harming the results, this thesis performs an ablation analysis. `CoSwitchMap v2` has two differences with `CoSwitchMap`: it uses a frequency-based extraction of code-switching pairs instead of a script-based one, and it uses a simplified mapping learning method using Procrustes instead of a modified skip-gram. This ablation analysis thus compares `CoSwitchMap v2` with three baselines:

- removing the frequency-based extraction: and going back to the script-based one of `CoSwitchMap`
- removing the simplified mapping learning: and going back to the modified skip-gram of `CoSwitchMap`
- removing both: which is `CoSwitchMap`

Results are reported in Table 3.2. They show that the proposed modifications are note harming the results. There is no significant differences between the four combinations of ways to extract code-switching pairs and ways to learn the initial mapping. Eventually this shows that `CoSwitchMap` can be replaced by `CoSwitchMap v2`. This allows to generalize the experiment to more pairs of languages.

3.2.4 Results of `CoSwitchMap v2`

`CoSwitchMap v2` is compared to other baselines on all the language pairs in which those baselines were previously evaluated. Results are shown in Table 3.3. For pairs of languages that are easy to align like English-French (fr) or English-Italian (it), `CoSwitchMap v2` matches or outperforms the other unsupervised methods, providing better results than the supervised baseline. For other pairs of languages, while fully unsupervised methods sometimes dramatically fail to reach near-zero accuracy, it is not the case of `CoSwitchMap v2` and the weakly-supervised Vecmap identical. There is however one outlier language for which `CoSwitchMap v2` fails: Tamil (ta), which this thesis cannot explain.

For all other languages, `CoSwitchMap v2` is competitive with Vecmap identical. Unfortunately, Chinese (zh) and Italian (it) are the only two languages for which `CoSwitchMap v2` significantly outperforms any other unsupervised baseline. On other hard-to-align pairs of languages, `CoSwitchMap v2` and Vecmap identical obtain almost the same results (ar, af, bs and th). But for the two languages that are the hardest to align to English, Tagalog (tl) and Tamil (ta), Vecmap identical significantly outperforms the proposed method.

method	fr	it	hu	ja	ru	vi	fi
MUSE	$82.2_{\pm 0.2}$	$77.5_{\pm 0.4}$	$49.9_{\pm 1.8}$	$0.0_{\pm 0.0}$	$41.7_{\pm 2.9}$	$0.0_{\pm 0.0}$	$41.9_{\pm 2.4}$
WP	$81.0_{\pm 0.2}$	$74.3_{\pm 1.0}$	$43.5_{\pm 1.7}$	$3.8_{\pm 6.8}$	$36.9_{\pm 1.4}$	$0.0_{\pm 0.0}$	$0.2_{\pm 0.2}$
VecMap	$\mathbf{82.4}_{\pm 0.1}$	$79.0_{\pm 0.0}$	$56.1_{\pm 0.2}$	$11.2_{\pm 19.1}$	$\mathbf{49.1}_{\pm 0.4}$	$0.7_{\pm 0.4}$	$\mathbf{50.3}_{\pm 0.2}$
Vecmap w/ MUSE	$\mathbf{82.4}_{\pm 0.0}$	$\underline{79.1}_{\pm 0.1}$	$56.5_{\pm 0.4}$	$0.0_{\pm 0.0}$	$47.5_{\pm 2.5}$	$9.6_{\pm 19.1}$	$49.5_{\pm 0.6}$
Vecmap w/ WP	$\underline{82.3}_{\pm 0.0}$	$79.0_{\pm 0.1}$	$56.7_{\pm 0.2}$	$20.3_{\pm 23.1}$	$44.8_{\pm 3.3}$	$0.2_{\pm 0.1}$	$0.0_{\pm 0.1}$
Vecmap identical	$\underline{82.3}_{\pm 0.1}$	$79.0_{\pm 0.1}$	$\mathbf{57.0}_{\pm 0.2}$	$\mathbf{48.4}_{\pm 0.7}$	$\underline{48.9}_{\pm 0.2}$	$\mathbf{48.6}_{\pm 0.6}$	$\underline{50.2}_{\pm 0.2}$
`CoSwitchMap v2`	$\mathbf{82.4}_{\pm 0.1}$	$\mathbf{79.2}_{\pm 0.1}$	$56.3_{\pm 0.4}$	$\underline{48.3}_{\pm 0.5}$	$\underline{48.7}_{\pm 0.2}$	$47.7_{\pm 0.6}$	$49.2_{\pm 0.6}$
supervised w/ self-learning	$82.4_{\pm 0.0}$	$79.0_{\pm 0.1}$	$57.1_{\pm 0.2}$	$48.7_{\pm 0.2}$	$49.1_{\pm 0.2}$	$47.9_{\pm 0.4}$	$50.3_{\pm 0.2}$
supervised	81.6	78.8	57.2	52.8	52.7	49.8	49.2

method	zh	ar	af	bs	th	tl	ta
MUSE	$0.0_{\pm 3.3}$	$30.9_{\pm 3.3}$	$4.5_{\pm 8.9}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	-*	$0.0_{\pm 0.0}$
WP	$0.6_{\pm 0.8}$	$10.7_{\pm 9.9}$	$23.5_{\pm 1.1}$	$0.5_{\pm 0.6}$	$0.0_{\pm 0.0}$	$3.0_{\pm 3.0}$	$0.0_{\pm 0.0}$
VecMap	$0.0_{\pm 0.0}$	$36.1_{\pm 1.8}$	$33.7_{\pm 1.0}$	$0.1_{\pm 0.2}$	$0.0_{\pm 0.0}$	$0.1_{\pm 0.1}$	$0.1_{\pm 0.1}$
Vecmap w/ MUSE	$6.5_{\pm 12.6}$	$\underline{39.5}_{\pm 0.7}$	$7.2_{\pm 14.0}$	$0.1_{\pm 0.2}$	$0.0_{\pm 0.0}$	-*	$3.0_{\pm 5.9}$
Vecmap w/ WP	$0.0_{\pm 0.1}$	$31.9_{\pm 15.9}$	$\underline{34.7}_{\pm 0.4}$	$0.2_{\pm 0.1}$	$0.0_{\pm 0.0}$	$8.7_{\pm 6.8}$	$0.0_{\pm 0.0}$
Vecmap identical	$36.8_{\pm 0.8}$	$\mathbf{39.8}_{\pm 0.3}$	$\mathbf{34.9}_{\pm 0.4}$	$\mathbf{28.3}_{\pm 0.8}$	$\underline{24.0}_{\pm 0.5}$	$\mathbf{19.4}_{\pm 0.4}$	$\mathbf{17.8}_{\pm 0.3}$
`CoSwitchMap v2`	$\mathbf{37.2}_{\pm 0.3}$	$\underline{39.5}_{\pm 0.3}$	$\mathbf{34.9}_{\pm 0.3}$	$\mathbf{28.3}_{\pm 0.6}$	$\mathbf{24.2}_{\pm 0.3}$	$18.8_{\pm 0.5}$	$0.0_{\pm 0.0}$
supervised w/ self-learning	$38.1_{\pm 0.4}$	$39.5_{\pm 0.1}$	$34.5_{\pm 0.4}$	$28.5_{\pm 0.3}$	$24.5_{\pm 0.2}$	$19.1_{\pm 0.6}$	$18.3_{\pm 0.5}$
supervised	43.3	43.0	34.8	28.9	25.1	22.8	20.2

Table 3.3 – Results of `CoSwitchMap v2` are reported alongside results of the other baselines.

While the results of `CoSwitchMap` and `CoSwitchMap v2` on Chinese seems promising, there are not any other language yet, except maybe Italian, for which using code-switching provide better results than simply using a dictionary of identical words as a training signal. But as stated earlier, using a dictionary of identical words probably relies at least partially, indirectly on code-switching, since code-switching will make words from one language appear in the vocabulary of the other.

`CoSwitchMap v2` can be useful for very specific languages, or at least Chinese, but in a fully unsupervised setting, this could not have been known in advance. However, its results are competitive with all unsupervised baselines, which shows that code-switching is a valuable cross-lingual signal, even when the method to extract it is based on word frequency.

Chapter 4

Detailed results on MedNERF

Detailed results are shown in the following tables:

- Summary of results for cross-lingual adaptation, with pre-selected and best pairs of translation and alignment models: Table 4.1 for French and 4.2 for German;
- CTL and `translate-train` with multilingual models: Table 4.3 (fr) and 4.4 (de);
- `translate-train` with language- and domain-specific models: Table 4.5 and 4.6;
- `translate-test` with multilingual language models: Table 4.7 and 4.8;
- `translate-test` with PubmedBERT: Table 4.9 and 4.10;
- Breakdown of the results class-by-class in French for multilingual models: Table 4.11.

model	pre-selected	best
translate-train		
distilmBERT	$66.5_{\pm 1.9}$	$69.2_{\pm 1.2}$
XLM-R Base	$74.2_{\pm 0.9}$	$78.6_{\pm 0.5}$
XLM-R Large	$76.5_{\pm 0.7}$	$78.0_{\pm 0.5}$
CamemBERT	$73.5_{\pm 1.5}$	$76.7_{\pm 0.9}$
DrBERT	$70.7_{\pm 1.3}$	$73.5_{\pm 1.4}$
DrBERT Pubmed	$76.1_{\pm 1.3}$	$78.8_{\pm 1.4}$
translate-test		
distilmBERT	$69.2_{\pm 1.4}$	$69.7_{\pm 1.5}$
XLM-R Base	$74.2_{\pm 1.6}$	$74.4_{\pm 1.3}$
XLM-R Large	$75.3_{\pm 0.9}$	$75.3_{\pm 0.9}$
PubmedBERT	$73.3_{\pm 1.3}$	$73.5_{\pm 1.2}$
CTL*		
distilmBERT	$65.9_{\pm 3.3}$	$65.9_{\pm 3.3}$
+ realigned	$66.4_{\pm 1.4}$	$66.4_{\pm 1.4}$
XLM-R Base	$\mathbf{79.1}_{\pm 0.8}$	$\mathbf{79.1}_{\pm 0.8}$
+ realigned	$76.7_{\pm 0.7}$	$76.7_{\pm 0.7}$
XLM-R Large	$77.9_{\pm 1.7}$	$77.9_{\pm 1.7}$
+ realigned	$78.8_{\pm 1.6}$	$78.8_{\pm 1.6}$

Table 4.1 – Summary of results for cross-lingual adaptation to French.
*results are reported twice as there is no pre-selection process

model	pre-selected	best
translate-train		
distilmBERT	$68.3_{\pm 1.3}$	$69.2_{\pm 1.2}$
XLM-R Base	$73.7_{\pm 0.9}$	$74.8_{\pm 1.0}$
XLM-R Large	$77.4_{\pm 1.3}$	$\mathbf{79.4}_{\pm 1.3}$
GottBERT	$75.5_{\pm 1.4}$	$76.6_{\pm 0.8}$
MedBERT.de	$72.7_{\pm 0.5}$	$75.0_{\pm 1.6}$
translate-test		
distilmBERT	$68.3_{\pm 1.8}$	$68.3_{\pm 1.8}$
XLM-R Base	$72.7_{\pm 0.8}$	$72.7_{\pm 0.8}$
XLM-R Large	$76.1_{\pm 2.8}$	$76.1_{\pm 2.8}$
PubmedBERT	$72.6_{\pm 1.5}$	$73.3_{\pm 1.7}$
CTL*		
distilmBERT	$64.6_{\pm 2.4}$	$64.6_{\pm 2.4}$
+ realigned	$67.9_{\pm 1.5}$	$67.9_{\pm 1.5}$
XLM-R Base	$72.2_{\pm 0.7}$	$72.2_{\pm 0.7}$
+ realigned	$75.8_{\pm 1.3}$	$75.8_{\pm 1.3}$
XLM-R Large	$\mathbf{78.5}_{\pm 0.4}$	$78.5_{\pm 0.4}$
+ realigned	$78.3_{\pm 1.6}$	$78.3_{\pm 1.6}$

Table 4.2 – Summary of results for cross-lingual adaptation to German.
*results reported twice as there is no pre-selection process

translation	aligner	precision	recall	micro-f1	macro-f1
distilmBERT					
Opus	FastAlign	$67.8_{\pm 2.1}$	$68.7_{\pm 1.5}$	$68.3_{\pm 1.6}$	$70.7_{\pm 1.4}$
Opus	AWESOME w/o co	$67.1_{\pm 1.0}$	$68.7_{\pm 1.3}$	$67.9_{\pm 0.3}$	$70.4_{\pm 0.3}$
Opus	AWESOME w/ co	$\mathbf{68.2}_{\pm 1.6}$	$70.1_{\pm 1.1}$	$\mathbf{69.2}_{\pm 1.2}$	$\mathbf{71.7}_{\pm 1.1}$
Opus	AWESOME ft w/o co	$65.0_{\pm 0.8}$	$67.4_{\pm 1.8}$	$66.2_{\pm 1.0}$	$68.8_{\pm 0.8}$
Opus	AWESOME ft w/ co	$65.2_{\pm 1.2}$	$68.4_{\pm 1.8}$	$66.8_{\pm 1.3}$	$69.4_{\pm 1.1}$
Opus	AWESOME pt+ft w/o co	$66.5_{\pm 1.3}$	$69.1_{\pm 1.2}$	$67.8_{\pm 1.0}$	$70.2_{\pm 0.9}$
Opus	AWESOME pt+ft w/ co	$66.6_{\pm 1.2}$	$\mathbf{70.2}_{\pm 1.8}$	$68.3_{\pm 1.1}$	$70.8_{\pm 1.1}$
Opus ft	FastAlign	$66.0_{\pm 1.1}$	$69.5_{\pm 0.4}$	$67.7_{\pm 0.5}$	$70.0_{\pm 0.5}$
Opus ft	AWESOME w/o co	$65.2_{\pm 1.9}$	$67.8_{\pm 1.9}$	$66.5_{\pm 1.9}$	$69.1_{\pm 1.7}$
Opus ft	AWESOME w/ co	$65.7_{\pm 1.2}$	$68.3_{\pm 1.9}$	$66.9_{\pm 1.4}$	$69.9_{\pm 1.2}$
Opus ft	AWESOME ft w/o co	$64.6_{\pm 0.8}$	$67.4_{\pm 1.0}$	$65.9_{\pm 0.4}$	$68.6_{\pm 0.5}$
Opus ft	AWESOME ft w/ co	$64.9_{\pm 0.7}$	$68.8_{\pm 1.5}$	$66.8_{\pm 1.1}$	$69.4_{\pm 1.0}$
Opus ft	AWESOME pt+ft w/o co	$64.5_{\pm 1.1}$	$68.1_{\pm 1.4}$	$66.2_{\pm 1.0}$	$69.0_{\pm 1.0}$
Opus ft	AWESOME pt+ft w/ co	$63.5_{\pm 1.0}$	$68.3_{\pm 1.1}$	$65.8_{\pm 0.9}$	$68.7_{\pm 1.0}$
Cross-lingual Transfer		$64.9_{\pm 2.5}$	$66.8_{\pm 4.1}$	$65.9_{\pm 3.3}$	$68.2_{\pm 3.2}$
Cross-lingual Transfer with realignment		$68.1_{\pm 2.2}$	$64.9_{\pm 1.0}$	$66.4_{\pm 1.4}$	$67.4_{\pm 1.6}$
XLM-R Base					
Opus	FastAlign	$77.4_{\pm 0.9}$	$79.1_{\pm 0.7}$	$78.2_{\pm 0.8}$	$79.8_{\pm 0.6}$
Opus	AWESOME w/o co	$75.4_{\pm 2.4}$	$77.3_{\pm 1.6}$	$76.4_{\pm 2.0}$	$78.6_{\pm 1.6}$
Opus	AWESOME w/ co	$76.0_{\pm 1.0}$	$77.6_{\pm 1.6}$	$76.8_{\pm 1.2}$	$78.8_{\pm 1.2}$
Opus	AWESOME ft w/o co	$73.8_{\pm 1.7}$	$75.4_{\pm 1.6}$	$74.6_{\pm 1.6}$	$76.8_{\pm 1.4}$
Opus	AWESOME ft w/ co	$75.0_{\pm 0.8}$	$76.9_{\pm 0.6}$	$76.0_{\pm 0.5}$	$78.2_{\pm 0.6}$
Opus	AWESOME pt+ft w/o co	$74.8_{\pm 1.5}$	$76.8_{\pm 2.0}$	$75.8_{\pm 1.7}$	$78.0_{\pm 1.7}$
Opus	AWESOME pt+ft w/ co	$75.1_{\pm 1.3}$	$76.8_{\pm 1.6}$	$76.0_{\pm 1.4}$	$78.2_{\pm 1.2}$
Opus ft	FastAlign	$77.9_{\pm 0.2}$	$79.4_{\pm 1.0}$	$78.6_{\pm 0.5}$	$80.3_{\pm 0.3}$
Opus ft	AWESOME w/o co	$72.9_{\pm 1.4}$	$75.6_{\pm 0.8}$	$74.2_{\pm 0.9}$	$76.7_{\pm 0.7}$
Opus ft	AWESOME w/ co	$74.6_{\pm 1.0}$	$76.6_{\pm 0.8}$	$75.6_{\pm 0.9}$	$77.9_{\pm 0.7}$
Opus ft	AWESOME ft w/o co	$73.4_{\pm 1.7}$	$75.6_{\pm 1.6}$	$74.5_{\pm 1.6}$	$77.0_{\pm 1.4}$
Opus ft	AWESOME ft w/ co	$74.2_{\pm 1.9}$	$76.7_{\pm 2.2}$	$75.5_{\pm 2.0}$	$78.0_{\pm 1.7}$
Opus ft	AWESOME pt+ft w/o co	$75.1_{\pm 1.0}$	$77.5_{\pm 1.1}$	$76.3_{\pm 1.0}$	$78.5_{\pm 0.9}$
Opus ft	AWESOME pt+ft w/ co	$75.1_{\pm 1.9}$	$77.6_{\pm 1.7}$	$76.3_{\pm 1.8}$	$78.7_{\pm 1.5}$
Cross-lingual Transfer		$\mathbf{78.7}_{\pm 1.8}$	$\mathbf{79.6}_{\pm 0.5}$	$\mathbf{79.1}_{\pm 0.8}$	$\mathbf{80.9}_{\pm 0.9}$
Cross-lingual Transfer with realignment		$76.9_{\pm 1.4}$	$76.6_{\pm 0.2}$	$76.7_{\pm 0.7}$	$78.9_{\pm 0.8}$
XLM-R Large					
Opus	FastAlign	$78.8_{\pm 0.7}$	$77.2_{\pm 0.6}$	$78.0_{\pm 0.5}$	$79.8_{\pm 0.5}$
Opus	AWESOME w/o co	$76.9_{\pm 1.1}$	$76.1_{\pm 1.8}$	$76.5_{\pm 1.2}$	$78.7_{\pm 1.1}$
Opus	AWESOME w/ co	$76.5_{\pm 1.2}$	$75.3_{\pm 1.5}$	$75.9_{\pm 1.3}$	$78.2_{\pm 1.1}$
Opus	AWESOME ft w/o co	$74.6_{\pm 1.0}$	$73.9_{\pm 0.4}$	$74.2_{\pm 0.6}$	$76.8_{\pm 0.5}$
Opus	AWESOME ft w/ co	$74.9_{\pm 0.3}$	$75.7_{\pm 0.8}$	$75.3_{\pm 0.3}$	$78.1_{\pm 0.5}$
Opus	AWESOME pt+ft w/o co	$75.8_{\pm 1.4}$	$75.0_{\pm 1.0}$	$75.4_{\pm 1.0}$	$78.0_{\pm 0.7}$
Opus	AWESOME pt+ft w/ co	$75.7_{\pm 2.2}$	$76.2_{\pm 1.3}$	$75.9_{\pm 1.7}$	$78.3_{\pm 1.8}$
Opus ft	FastAlign	$76.2_{\pm 2.0}$	$76.9_{\pm 2.5}$	$76.6_{\pm 2.1}$	$78.3_{\pm 2.0}$
Opus ft	AWESOME w/o co	$76.1_{\pm 1.0}$	$76.9_{\pm 0.8}$	$76.5_{\pm 0.7}$	$78.9_{\pm 0.5}$
Opus ft	AWESOME w/ co	$74.6_{\pm 1.5}$	$76.0_{\pm 1.3}$	$75.3_{\pm 0.9}$	$77.9_{\pm 0.6}$
Opus ft	AWESOME ft w/o co	$74.1_{\pm 1.4}$	$75.5_{\pm 0.4}$	$74.8_{\pm 0.8}$	$77.4_{\pm 0.8}$
Opus ft	AWESOME ft w/ co	$75.5_{\pm 1.9}$	$75.6_{\pm 1.2}$	$75.5_{\pm 1.4}$	$78.1_{\pm 1.2}$
Opus ft	AWESOME pt+ft w/o co	$75.1_{\pm 0.5}$	$76.0_{\pm 1.2}$	$75.5_{\pm 0.6}$	$78.1_{\pm 0.5}$
Opus ft	AWESOME pt+ft w/ co	$75.0_{\pm 1.8}$	$75.8_{\pm 0.8}$	$75.4_{\pm 1.1}$	$77.7_{\pm 1.0}$
Cross-lingual Transfer		$78.2_{\pm 2.5}$	$77.6_{\pm 1.2}$	$77.9_{\pm 1.7}$	$80.0_{\pm 1.4}$
Cross-lingual Transfer with realignment		$\mathbf{79.7}_{\pm 1.6}$	$\mathbf{77.9}_{\pm 1.8}$	$\mathbf{78.8}_{\pm 1.6}$	$\mathbf{80.8}_{\pm 1.4}$

Table 4.3 – Cross-lingual Transfer and `translate-train` results in French for multilingual base models.

translation	aligner	precision	recall	micro-f1	macro-f1
distilmBERT					
FAIR	FastAlign	$69.0_{\pm 1.2}$	$63.9_{\pm 0.5}$	$66.3_{\pm 0.8}$	$44.2_{\pm 2.4}$
FAIR	AWESOME w/o co	$68.1_{\pm 1.6}$	$66.1_{\pm 1.3}$	$67.1_{\pm 1.3}$	$50.5_{\pm 3.0}$
FAIR	AWESOME w/ co	$68.0_{\pm 2.0}$	$\mathbf{67.4}_{\pm 1.4}$	$67.7_{\pm 1.0}$	$50.6_{\pm 3.7}$
FAIR	AWESOME ft w/o co	$68.5_{\pm 3.6}$	$65.0_{\pm 2.2}$	$66.6_{\pm 1.2}$	$48.3_{\pm 4.0}$
FAIR	AWESOME ft w/ co	$67.6_{\pm 2.4}$	$66.1_{\pm 3.5}$	$66.7_{\pm 1.4}$	$46.4_{\pm 5.1}$
FAIR	AWESOME pt+ft w/o co	$69.6_{\pm 2.7}$	$65.9_{\pm 2.8}$	$67.6_{\pm 1.1}$	$46.8_{\pm 6.4}$
FAIR	AWESOME pt+ft w/ co	$68.8_{\pm 2.6}$	$66.6_{\pm 3.7}$	$67.5_{\pm 1.4}$	$49.2_{\pm 6.0}$
FAIR ft	FastAlign	$70.5_{\pm 1.9}$	$65.2_{\pm 1.4}$	$67.7_{\pm 1.3}$	$\mathbf{52.2}_{\pm 1.5}$
FAIR ft	AWESOME w/o co	$69.3_{\pm 2.0}$	$66.9_{\pm 1.6}$	$68.0_{\pm 1.1}$	$48.0_{\pm 3.6}$
FAIR ft	AWESOME w/ co	$68.6_{\pm 2.8}$	$66.4_{\pm 2.0}$	$67.4_{\pm 1.5}$	$47.4_{\pm 4.0}$
FAIR ft	AWESOME ft w/o co	$70.9_{\pm 2.7}$	$67.2_{\pm 2.1}$	$69.0_{\pm 1.9}$	$49.8_{\pm 3.7}$
FAIR ft	AWESOME ft w/ co	$71.4_{\pm 2.0}$	$67.2_{\pm 1.8}$	$\mathbf{69.2}_{\pm 1.2}$	$49.0_{\pm 4.3}$
FAIR ft	AWESOME pt+ft w/o co	$69.3_{\pm 1.8}$	$\mathbf{67.4}_{\pm 1.9}$	$68.3_{\pm 1.3}$	$49.2_{\pm 3.2}$
FAIR ft	AWESOME pt+ft w/ co	$70.4_{\pm 1.9}$	$67.1_{\pm 1.6}$	$68.7_{\pm 1.2}$	$49.5_{\pm 3.1}$
Cross-lingual Transfer		$62.8_{\pm 2.9}$	$66.4_{\pm 2.0}$	$64.6_{\pm 2.4}$	$46.3_{\pm 3.1}$
Cross-lingual Transfer with realignment		$\mathbf{72.0}_{\pm 2.3}$	$64.2_{\pm 1.4}$	$67.9_{\pm 1.5}$	$46.5_{\pm 5.2}$
XLM-R Base					
FAIR	FastAlign	$74.0_{\pm 2.6}$	$71.6_{\pm 1.4}$	$72.8_{\pm 1.6}$	$55.5_{\pm 5.7}$
FAIR	AWESOME w/o co	$73.3_{\pm 1.8}$	$72.9_{\pm 0.8}$	$73.1_{\pm 1.3}$	$53.2_{\pm 3.8}$
FAIR	AWESOME w/ co	$73.9_{\pm 2.8}$	$72.8_{\pm 2.2}$	$73.3_{\pm 2.3}$	$53.1_{\pm 4.4}$
FAIR	AWESOME ft w/o co	$74.1_{\pm 1.1}$	$74.1_{\pm 1.1}$	$74.1_{\pm 1.1}$	$57.5_{\pm 4.3}$
FAIR	AWESOME ft w/ co	$75.4_{\pm 2.3}$	$74.1_{\pm 1.4}$	$74.8_{\pm 1.8}$	$58.0_{\pm 4.2}$
FAIR	AWESOME pt+ft w/o co	$74.6_{\pm 1.8}$	$73.6_{\pm 1.1}$	$74.1_{\pm 1.3}$	$56.3_{\pm 2.2}$
FAIR	AWESOME pt+ft w/ co	$74.9_{\pm 1.9}$	$73.8_{\pm 1.0}$	$74.4_{\pm 1.4}$	$57.5_{\pm 3.3}$
FAIR ft	FastAlign	$74.7_{\pm 2.2}$	$72.1_{\pm 0.6}$	$73.3_{\pm 1.3}$	$54.2_{\pm 4.6}$
FAIR ft	AWESOME w/o co	$76.8_{\pm 1.6}$	$72.8_{\pm 1.0}$	$74.7_{\pm 1.0}$	$53.0_{\pm 2.7}$
FAIR ft	AWESOME w/ co	$75.9_{\pm 1.7}$	$\mathbf{73.8}_{\pm 0.6}$	$74.8_{\pm 1.0}$	$57.5_{\pm 4.5}$
FAIR ft	AWESOME ft w/o co	$77.0_{\pm 1.5}$	$72.1_{\pm 0.6}$	$74.5_{\pm 0.8}$	$51.0_{\pm 0.9}$
FAIR ft	AWESOME ft w/ co	$76.2_{\pm 1.2}$	$72.1_{\pm 1.3}$	$74.1_{\pm 1.1}$	$50.9_{\pm 1.0}$
FAIR ft	AWESOME pt+ft w/o co	$75.5_{\pm 1.2}$	$71.9_{\pm 1.1}$	$73.7_{\pm 0.9}$	$52.1_{\pm 3.5}$
FAIR ft	AWESOME pt+ft w/ co	$75.5_{\pm 0.9}$	$72.6_{\pm 1.1}$	$74.0_{\pm 1.0}$	$52.6_{\pm 3.8}$
Cross-lingual Transfer		$71.1_{\pm 1.1}$	$73.3_{\pm 1.0}$	$72.2_{\pm 0.7}$	$55.1_{\pm 5.7}$
Cross-lingual Transfer with realignment		$\mathbf{78.2}_{\pm 1.8}$	$73.5_{\pm 1.9}$	$\mathbf{75.8}_{\pm 1.3}$	$\mathbf{58.1}_{\pm 4.9}$
XLM-R Large					
FAIR	FastAlign	$77.7_{\pm 3.6}$	$75.1_{\pm 2.3}$	$76.4_{\pm 2.7}$	$65.8_{\pm 2.8}$
FAIR	AWESOME w/o co	$80.1_{\pm 1.1}$	$77.5_{\pm 0.6}$	$78.7_{\pm 0.5}$	$65.0_{\pm 3.1}$
FAIR	AWESOME w/ co	$79.9_{\pm 1.3}$	$77.0_{\pm 2.4}$	$78.4_{\pm 1.7}$	$64.4_{\pm 2.2}$
FAIR	AWESOME ft w/o co	$80.7_{\pm 1.6}$	$77.3_{\pm 0.9}$	$79.0_{\pm 0.6}$	$65.8_{\pm 0.6}$
FAIR	AWESOME ft w/ co	$80.0_{\pm 1.7}$	$\mathbf{78.5}_{\pm 1.1}$	$79.2_{\pm 1.1}$	$66.3_{\pm 1.1}$
FAIR	AWESOME pt+ft w/o co	$79.3_{\pm 0.8}$	$77.3_{\pm 1.1}$	$78.3_{\pm 0.7}$	$64.5_{\pm 2.0}$
FAIR	AWESOME pt+ft w/ co	$79.1_{\pm 1.8}$	$75.6_{\pm 1.2}$	$77.3_{\pm 0.8}$	$63.7_{\pm 1.6}$
FAIR ft	FastAlign	$78.5_{\pm 2.5}$	$75.1_{\pm 1.6}$	$76.7_{\pm 1.7}$	$61.7_{\pm 1.9}$
FAIR ft	AWESOME w/o co	$\mathbf{83.2}_{\pm 2.8}$	$76.0_{\pm 0.9}$	$\mathbf{79.4}_{\pm 1.6}$	$64.3_{\pm 2.2}$
FAIR ft	AWESOME w/ co	$83.0_{\pm 1.0}$	$76.1_{\pm 1.7}$	$\mathbf{79.4}_{\pm 1.3}$	$64.7_{\pm 1.3}$
FAIR ft	AWESOME ft w/o co	$80.0_{\pm 1.5}$	$76.0_{\pm 1.5}$	$77.9_{\pm 1.2}$	$64.9_{\pm 0.9}$
FAIR ft	AWESOME ft w/ co	$81.7_{\pm 2.0}$	$75.1_{\pm 1.7}$	$78.3_{\pm 1.8}$	$64.8_{\pm 1.6}$
FAIR ft	AWESOME pt+ft w/o co	$80.3_{\pm 1.9}$	$74.6_{\pm 1.6}$	$77.4_{\pm 1.3}$	$62.9_{\pm 3.2}$
FAIR ft	AWESOME pt+ft w/ co	$80.2_{\pm 0.9}$	$75.3_{\pm 2.0}$	$77.6_{\pm 1.1}$	$63.0_{\pm 2.9}$
Cross-lingual Transfer		$81.2_{\pm 1.2}$	$76.0_{\pm 0.9}$	$78.5_{\pm 0.4}$	$64.9_{\pm 2.5}$
Cross-lingual Transfer with realignment		$80.7_{\pm 2.1}$	$76.0_{\pm 1.4}$	$78.3_{\pm 1.6}$	$\mathbf{66.8}_{\pm 1.8}$

Table 4.4 – Cross-lingual Transfer and `translate-train` results in German for multilingual base models.

translation	aligner	precision	recall	mirco-f1	macro-f1
CamemBERT Base					
Opus	FastAlign	**74.9**$_{\pm 1.1}$	**78.5**$_{\pm 0.9}$	**76.7**$_{\pm 0.9}$	**78.7**$_{\pm 1.0}$
Opus	AWESOME w/o co	73.2$_{\pm 1.2}$	77.7$_{\pm 1.3}$	75.4$_{\pm 1.2}$	77.9$_{\pm 1.2}$
Opus	AWESOME w/ co	74.4$_{\pm 0.8}$	77.5$_{\pm 0.8}$	75.9$_{\pm 0.8}$	78.1$_{\pm 1.0}$
Opus	AWESOME ft w/o co	71.9$_{\pm 1.1}$	76.5$_{\pm 1.1}$	74.1$_{\pm 1.1}$	76.7$_{\pm 1.0}$
Opus	AWESOME ft w/ co	72.3$_{\pm 2.0}$	77.4$_{\pm 1.3}$	74.8$_{\pm 1.7}$	77.3$_{\pm 1.3}$
Opus	AWESOME pt+ft w/o co	74.0$_{\pm 1.4}$	77.8$_{\pm 1.5}$	75.9$_{\pm 1.4}$	78.2$_{\pm 1.3}$
Opus	AWESOME pt+ft w/ co	73.3$_{\pm 1.0}$	77.9$_{\pm 1.0}$	75.5$_{\pm 0.9}$	77.8$_{\pm 1.0}$
Opus ft	FastAlign	74.2$_{\pm 2.1}$	78.4$_{\pm 1.3}$	76.2$_{\pm 1.7}$	78.3$_{\pm 1.5}$
Opus ft	AWESOME w/o co	71.0$_{\pm 1.6}$	76.2$_{\pm 1.5}$	73.5$_{\pm 1.5}$	76.1$_{\pm 1.3}$
Opus ft	AWESOME w/ co	72.0$_{\pm 1.8}$	77.4$_{\pm 1.7}$	74.6$_{\pm 1.7}$	77.2$_{\pm 1.6}$
Opus ft	AWESOME ft w/o co	70.7$_{\pm 1.8}$	75.0$_{\pm 1.4}$	72.8$_{\pm 1.6}$	75.4$_{\pm 1.5}$
Opus ft	AWESOME ft w/ co	72.3$_{\pm 1.6}$	76.6$_{\pm 1.6}$	74.4$_{\pm 1.6}$	76.8$_{\pm 1.5}$
Opus ft	AWESOME pt+ft w/o co	72.2$_{\pm 2.4}$	77.1$_{\pm 1.5}$	74.6$_{\pm 1.9}$	76.9$_{\pm 1.7}$
Opus ft	AWESOME pt+ft w/ co	71.6$_{\pm 1.2}$	77.0$_{\pm 0.4}$	74.2$_{\pm 0.8}$	76.7$_{\pm 0.4}$
DrBERT 7GB					
Opus	FastAlign	70.9$_{\pm 2.4}$	72.4$_{\pm 2.1}$	71.7$_{\pm 2.1}$	73.2$_{\pm 2.1}$
Opus	AWESOME w/o co	69.5$_{\pm 1.4}$	71.3$_{\pm 1.5}$	70.4$_{\pm 1.3}$	72.3$_{\pm 1.6}$
Opus	AWESOME w/ co	69.5$_{\pm 1.7}$	71.7$_{\pm 0.7}$	70.6$_{\pm 0.8}$	72.6$_{\pm 0.7}$
Opus	AWESOME ft w/o co	69.3$_{\pm 1.1}$	71.7$_{\pm 0.7}$	70.4$_{\pm 0.8}$	72.7$_{\pm 0.7}$
Opus	AWESOME ft w/ co	68.1$_{\pm 0.8}$	70.3$_{\pm 1.5}$	69.2$_{\pm 0.7}$	71.3$_{\pm 0.8}$
Opus	AWESOME pt+ft w/o co	69.4$_{\pm 1.4}$	71.7$_{\pm 1.3}$	70.5$_{\pm 1.2}$	72.6$_{\pm 1.1}$
Opus	AWESOME pt+ft w/ co	70.0$_{\pm 1.4}$	71.4$_{\pm 1.0}$	70.7$_{\pm 0.6}$	72.7$_{\pm 0.5}$
Opus ft	FastAlign	**73.2**$_{\pm 1.9}$	**73.7**$_{\pm 1.5}$	**73.5**$_{\pm 1.4}$	**74.9**$_{\pm 1.4}$
Opus ft	AWESOME w/o co	69.6$_{\pm 1.6}$	71.7$_{\pm 1.1}$	70.7$_{\pm 1.3}$	72.7$_{\pm 1.1}$
Opus ft	AWESOME w/ co	70.5$_{\pm 1.6}$	71.9$_{\pm 1.2}$	71.2$_{\pm 1.2}$	73.4$_{\pm 1.1}$
Opus ft	AWESOME ft w/o co	69.1$_{\pm 1.5}$	70.6$_{\pm 1.1}$	69.8$_{\pm 0.5}$	71.8$_{\pm 0.3}$
Opus ft	AWESOME ft w/ co	70.8$_{\pm 1.5}$	72.6$_{\pm 1.3}$	71.6$_{\pm 1.0}$	73.7$_{\pm 0.8}$
Opus ft	AWESOME pt+ft w/o co	70.7$_{\pm 1.0}$	71.7$_{\pm 2.1}$	71.2$_{\pm 1.4}$	73.2$_{\pm 1.2}$
Opus ft	AWESOME pt+ft w/ co	70.1$_{\pm 0.8}$	70.6$_{\pm 1.7}$	70.4$_{\pm 1.1}$	72.4$_{\pm 1.2}$
DrBERT-PubMedBERT					
Opus	FastAlign	**76.2**$_{\pm 1.4}$	79.4$_{\pm 0.8}$	77.8$_{\pm 1.1}$	79.7$_{\pm 0.8}$
Opus	AWESOME w/o co	75.2$_{\pm 1.2}$	77.5$_{\pm 0.7}$	76.4$_{\pm 0.9}$	78.4$_{\pm 0.7}$
Opus	AWESOME w/ co	76.0$_{\pm 1.0}$	79.2$_{\pm 1.3}$	77.6$_{\pm 1.1}$	79.6$_{\pm 0.9}$
Opus	AWESOME ft w/o co	74.3$_{\pm 1.4}$	78.9$_{\pm 1.3}$	76.5$_{\pm 1.2}$	78.7$_{\pm 1.1}$
Opus	AWESOME ft w/ co	74.0$_{\pm 1.2}$	77.8$_{\pm 1.1}$	75.9$_{\pm 1.1}$	78.2$_{\pm 1.0}$
Opus	AWESOME pt+ft w/o co	73.9$_{\pm 1.1}$	77.7$_{\pm 0.6}$	75.8$_{\pm 0.9}$	78.1$_{\pm 0.7}$
Opus	AWESOME pt+ft w/ co	75.2$_{\pm 0.4}$	79.1$_{\pm 0.8}$	77.1$_{\pm 0.5}$	79.0$_{\pm 0.4}$
Opus ft	FastAlign	**76.2**$_{\pm 1.8}$	**81.5**$_{\pm 1.0}$	**78.8**$_{\pm 1.4}$	**80.4**$_{\pm 1.3}$
Opus ft	AWESOME w/o co	73.7$_{\pm 1.4}$	78.7$_{\pm 1.2}$	76.1$_{\pm 1.3}$	78.4$_{\pm 1.0}$
Opus ft	AWESOME w/ co	75.4$_{\pm 1.2}$	81.2$_{\pm 0.8}$	78.2$_{\pm 0.8}$	80.3$_{\pm 0.7}$
Opus ft	AWESOME ft w/o co	74.9$_{\pm 0.9}$	80.7$_{\pm 0.7}$	77.7$_{\pm 0.8}$	79.7$_{\pm 0.5}$
Opus ft	AWESOME ft w/ co	74.8$_{\pm 1.3}$	79.7$_{\pm 0.8}$	77.2$_{\pm 1.1}$	79.2$_{\pm 0.8}$
Opus ft	AWESOME pt+ft w/o co	75.6$_{\pm 1.1}$	79.3$_{\pm 1.5}$	77.4$_{\pm 1.2}$	79.4$_{\pm 1.1}$
Opus ft	AWESOME pt+ft w/ co	75.5$_{\pm 1.2}$	80.3$_{\pm 1.3}$	77.8$_{\pm 1.2}$	79.8$_{\pm 1.0}$

Table 4.5 – `translate-train` results in French for domain and language-specific base models.

translation	aligner	precision	recall	micro-f1	macro-f1
GottBERT					
FAIR	FastAlign	$75.9_{\pm 3.2}$	$70.3_{\pm 2.2}$	$73.0_{\pm 2.6}$	$54.8_{\pm 4.8}$
FAIR	AWESOME w/o co	$79.5_{\pm 1.6}$	$73.4_{\pm 2.0}$	$76.3_{\pm 1.7}$	$60.5_{\pm 4.6}$
FAIR	AWESOME w/ co	$77.9_{\pm 1.6}$	$72.9_{\pm 1.4}$	$75.3_{\pm 1.4}$	$57.2_{\pm 5.2}$
FAIR	AWESOME ft w/o co	$78.4_{\pm 2.8}$	$73.1_{\pm 2.4}$	$75.7_{\pm 2.4}$	$57.9_{\pm 5.4}$
FAIR	AWESOME ft w/ co	$77.9_{\pm 1.8}$	$72.6_{\pm 2.3}$	$75.1_{\pm 1.8}$	$53.4_{\pm 2.7}$
FAIR	AWESOME pt+ft w/o co	$79.0_{\pm 1.8}$	$\mathbf{74.1}_{\pm 1.6}$	$76.5_{\pm 1.6}$	$\mathbf{60.8}_{\pm 3.5}$
FAIR	AWESOME pt+ft w/ co	$77.9_{\pm 2.8}$	$73.6_{\pm 2.5}$	$75.7_{\pm 2.6}$	$57.6_{\pm 4.4}$
FAIR ft	FastAlign	$76.6_{\pm 3.2}$	$70.1_{\pm 1.9}$	$73.2_{\pm 2.4}$	$53.7_{\pm 4.5}$
FAIR ft	AWESOME w/o co	$\mathbf{80.2}_{\pm 1.1}$	$73.3_{\pm 1.0}$	$\mathbf{76.6}_{\pm 0.8}$	$58.7_{\pm 6.1}$
FAIR ft	AWESOME w/ co	$79.2_{\pm 0.8}$	$73.1_{\pm 1.1}$	$76.0_{\pm 0.7}$	$58.8_{\pm 2.4}$
FAIR ft	AWESOME ft w/o co	$78.5_{\pm 0.7}$	$72.4_{\pm 1.0}$	$75.3_{\pm 0.8}$	$56.1_{\pm 6.6}$
FAIR ft	AWESOME ft w/ co	$78.8_{\pm 2.3}$	$72.3_{\pm 1.9}$	$75.4_{\pm 1.8}$	$55.2_{\pm 6.4}$
FAIR ft	AWESOME pt+ft w/o co	$78.6_{\pm 1.6}$	$72.6_{\pm 1.6}$	$75.5_{\pm 1.4}$	$55.5_{\pm 6.4}$
FAIR ft	AWESOME pt+ft w/ co	$79.4_{\pm 1.8}$	$72.3_{\pm 0.8}$	$75.6_{\pm 1.1}$	$56.5_{\pm 3.2}$
medBERT.de					
FAIR	FastAlign	$75.1_{\pm 2.8}$	$69.2_{\pm 1.6}$	$72.0_{\pm 2.0}$	$56.7_{\pm 5.3}$
FAIR	AWESOME w/o co	$75.4_{\pm 1.6}$	$71.9_{\pm 1.8}$	$73.6_{\pm 1.2}$	$58.1_{\pm 5.0}$
FAIR	AWESOME w/ co	$\mathbf{77.0}_{\pm 3.5}$	$72.9_{\pm 1.6}$	$74.9_{\pm 2.4}$	$59.5_{\pm 5.9}$
FAIR	AWESOME ft w/o co	$76.4_{\pm 4.5}$	$70.9_{\pm 1.6}$	$73.5_{\pm 2.7}$	$56.2_{\pm 5.4}$
FAIR	AWESOME ft w/ co	$75.8_{\pm 4.0}$	$72.4_{\pm 2.5}$	$74.1_{\pm 3.1}$	$57.2_{\pm 6.4}$
FAIR	AWESOME pt+ft w/o co	$74.9_{\pm 3.9}$	$71.3_{\pm 1.6}$	$73.0_{\pm 2.4}$	$56.7_{\pm 5.5}$
FAIR	AWESOME pt+ft w/ co	$76.1_{\pm 4.1}$	$71.6_{\pm 1.9}$	$73.7_{\pm 2.6}$	$57.2_{\pm 5.7}$
FAIR ft	FastAlign	$72.6_{\pm 2.0}$	$68.9_{\pm 0.8}$	$70.7_{\pm 1.2}$	$60.7_{\pm 1.2}$
FAIR ft	AWESOME w/o co	$75.2_{\pm 2.0}$	$72.6_{\pm 0.9}$	$73.9_{\pm 1.0}$	$61.1_{\pm 1.9}$
FAIR ft	AWESOME w/ co	$76.4_{\pm 2.4}$	$\mathbf{73.6}_{\pm 0.9}$	$\mathbf{75.0}_{\pm 1.6}$	$62.2_{\pm 3.3}$
FAIR ft	AWESOME ft w/o co	$75.1_{\pm 3.6}$	$71.8_{\pm 2.8}$	$73.4_{\pm 3.1}$	$60.5_{\pm 3.5}$
FAIR ft	AWESOME ft w/ co	$75.3_{\pm 3.2}$	$71.9_{\pm 2.2}$	$73.6_{\pm 2.4}$	$58.6_{\pm 6.1}$
FAIR ft	AWESOME pt+ft w/o co	$74.1_{\pm 1.1}$	$71.4_{\pm 0.8}$	$72.7_{\pm 0.5}$	$57.8_{\pm 4.9}$
FAIR ft	AWESOME pt+ft w/ co	$75.5_{\pm 1.3}$	$72.3_{\pm 1.0}$	$73.8_{\pm 1.0}$	$\mathbf{62.6}_{\pm 1.2}$

Table 4.6 – `translate-train` results in German for domain and language-specific base models.

translation	aligner	precision	recall	micro-f1	macro-f1
distilmBERT					
Opus	FastAlign	$65.9_{\pm 1.8}$	$67.0_{\pm 1.5}$	$66.4_{\pm 1.5}$	$68.5_{\pm 1.3}$
Opus	AWESOME w/o co	$\mathbf{70.5}_{\pm 1.7}$	$68.8_{\pm 1.4}$	$69.6_{\pm 1.4}$	$71.8_{\pm 1.3}$
Opus	AWESOME w/ co	$\mathbf{70.5}_{\pm 1.7}$	$\mathbf{69.0}_{\pm 1.4}$	$\mathbf{69.7}_{\pm 1.5}$	$\mathbf{71.9}_{\pm 1.3}$
Opus	AWESOME ft w/o co	$70.0_{\pm 1.8}$	$68.8_{\pm 1.6}$	$69.4_{\pm 1.6}$	$71.6_{\pm 1.4}$
Opus	AWESOME ft w/ co	$69.7_{\pm 1.8}$	$68.8_{\pm 1.6}$	$69.2_{\pm 1.6}$	$71.5_{\pm 1.4}$
Opus	AWESOME pt+ft w/o co	$70.1_{\pm 1.7}$	$68.9_{\pm 1.5}$	$69.5_{\pm 1.5}$	$71.7_{\pm 1.4}$
Opus	AWESOME pt+ft w/ co	$69.0_{\pm 1.8}$	$68.8_{\pm 1.6}$	$68.9_{\pm 1.6}$	$71.3_{\pm 1.4}$
Opus ft	FastAlign	$63.2_{\pm 1.7}$	$66.6_{\pm 1.6}$	$64.8_{\pm 1.5}$	$66.7_{\pm 1.2}$
Opus ft	AWESOME w/o co	$69.9_{\pm 1.6}$	$68.4_{\pm 1.3}$	$69.2_{\pm 1.4}$	$71.4_{\pm 1.2}$
Opus ft	AWESOME w/ co	$69.2_{\pm 1.7}$	$68.7_{\pm 1.4}$	$68.9_{\pm 1.5}$	$71.3_{\pm 1.3}$
Opus ft	AWESOME ft w/o co	$69.4_{\pm 1.8}$	$68.6_{\pm 1.6}$	$69.0_{\pm 1.6}$	$71.2_{\pm 1.4}$
Opus ft	AWESOME ft w/ co	$69.1_{\pm 1.7}$	$68.6_{\pm 1.5}$	$68.9_{\pm 1.5}$	$71.1_{\pm 1.3}$
Opus ft	AWESOME pt+ft w/o co	$69.1_{\pm 1.7}$	$68.6_{\pm 1.5}$	$68.8_{\pm 1.5}$	$71.1_{\pm 1.3}$
Opus ft	AWESOME pt+ft w/ co	$67.3_{\pm 1.7}$	$68.6_{\pm 1.5}$	$67.9_{\pm 1.5}$	$70.5_{\pm 1.3}$
XLM-R Base					
Opus	FastAlign	$68.7_{\pm 1.7}$	$71.8_{\pm 1.2}$	$70.2_{\pm 1.4}$	$72.4_{\pm 1.2}$
Opus	AWESOME w/o co	$74.9_{\pm 1.5}$	$73.8_{\pm 1.2}$	$74.3_{\pm 1.3}$	$76.5_{\pm 1.1}$
Opus	AWESOME w/ co	$74.8_{\pm 1.6}$	$\mathbf{74.1}_{\pm 1.3}$	$\mathbf{74.4}_{\pm 1.3}$	$\mathbf{76.6}_{\pm 1.2}$
Opus	AWESOME ft w/o co	$74.6_{\pm 1.6}$	$73.8_{\pm 1.2}$	$74.2_{\pm 1.2}$	$76.3_{\pm 1.1}$
Opus	AWESOME ft w/ co	$74.1_{\pm 1.5}$	$73.8_{\pm 1.2}$	$74.0_{\pm 1.3}$	$76.1_{\pm 1.1}$
Opus	AWESOME pt+ft w/o co	$74.6_{\pm 1.6}$	$73.8_{\pm 1.2}$	$74.2_{\pm 1.2}$	$76.3_{\pm 1.1}$
Opus	AWESOME pt+ft w/ co	$73.4_{\pm 1.5}$	$73.8_{\pm 1.2}$	$73.6_{\pm 1.2}$	$75.9_{\pm 1.1}$
Opus ft	FastAlign	$67.7_{\pm 1.6}$	$71.7_{\pm 1.7}$	$69.6_{\pm 1.4}$	$71.4_{\pm 1.3}$
Opus ft	AWESOME w/o co	$\mathbf{75.4}_{\pm 1.7}$	$73.1_{\pm 1.7}$	$74.2_{\pm 1.6}$	$76.3_{\pm 1.5}$
Opus ft	AWESOME w/ co	$74.2_{\pm 1.7}$	$73.3_{\pm 1.7}$	$73.8_{\pm 1.6}$	$76.0_{\pm 1.6}$
Opus ft	AWESOME ft w/o co	$74.6_{\pm 1.8}$	$73.1_{\pm 1.7}$	$73.8_{\pm 1.6}$	$75.8_{\pm 1.5}$
Opus ft	AWESOME ft w/ co	$74.1_{\pm 1.8}$	$73.1_{\pm 1.7}$	$73.6_{\pm 1.6}$	$75.6_{\pm 1.5}$
Opus ft	AWESOME pt+ft w/o co	$74.3_{\pm 1.8}$	$73.2_{\pm 1.8}$	$73.7_{\pm 1.7}$	$75.7_{\pm 1.6}$
Opus ft	AWESOME pt+ft w/ co	$72.3_{\pm 1.7}$	$73.1_{\pm 1.7}$	$72.7_{\pm 1.6}$	$75.0_{\pm 1.5}$
XLM-R Large					
Opus	FastAlign	$69.6_{\pm 1.3}$	$71.0_{\pm 0.9}$	$70.3_{\pm 1.1}$	$72.7_{\pm 0.9}$
Opus	AWESOME w/o co	$75.9_{\pm 1.0}$	$73.4_{\pm 0.8}$	$74.7_{\pm 0.9}$	$77.0_{\pm 0.8}$
Opus	AWESOME w/ co	$76.0_{\pm 1.0}$	$\mathbf{73.7}_{\pm 0.7}$	$74.8_{\pm 0.8}$	$77.2_{\pm 0.8}$
Opus	AWESOME ft w/o co	$75.5_{\pm 1.0}$	$73.5_{\pm 0.8}$	$74.5_{\pm 0.9}$	$76.8_{\pm 0.8}$
Opus	AWESOME ft w/ co	$75.2_{\pm 1.1}$	$73.5_{\pm 0.8}$	$74.3_{\pm 0.9}$	$76.7_{\pm 0.8}$
Opus	AWESOME pt+ft w/o co	$75.6_{\pm 1.1}$	$73.5_{\pm 0.8}$	$74.5_{\pm 0.9}$	$76.8_{\pm 0.8}$
Opus	AWESOME pt+ft w/ co	$74.5_{\pm 1.1}$	$73.5_{\pm 0.8}$	$74.0_{\pm 0.9}$	$76.5_{\pm 0.8}$
Opus ft	FastAlign	$70.8_{\pm 1.4}$	$72.2_{\pm 0.5}$	$71.5_{\pm 1.0}$	$73.2_{\pm 0.9}$
Opus ft	AWESOME w/o co	$\mathbf{77.3}_{\pm 1.3}$	$73.3_{\pm 0.6}$	$\mathbf{75.3}_{\pm 0.9}$	$\mathbf{77.6}_{\pm 0.7}$
Opus ft	AWESOME w/ co	$76.3_{\pm 1.1}$	$73.6_{\pm 0.6}$	$75.0_{\pm 0.9}$	$77.4_{\pm 0.6}$
Opus ft	AWESOME ft w/o co	$76.4_{\pm 1.3}$	$73.4_{\pm 0.6}$	$74.9_{\pm 0.9}$	$77.1_{\pm 0.7}$
Opus ft	AWESOME ft w/ co	$76.1_{\pm 1.2}$	$73.4_{\pm 0.6}$	$74.7_{\pm 0.9}$	$77.0_{\pm 0.7}$
Opus ft	AWESOME pt+ft w/o co	$76.1_{\pm 1.2}$	$73.5_{\pm 0.6}$	$74.8_{\pm 0.8}$	$77.0_{\pm 0.7}$
Opus ft	AWESOME pt+ft w/ co	$74.1_{\pm 1.1}$	$73.4_{\pm 0.6}$	$73.8_{\pm 0.8}$	$76.3_{\pm 0.6}$

Table 4.7 – Full results for the `translate-test` approach in French with multilingual language models.

translation	aligner	precision	recall	micro-f1	macro-f1
distilmBERT					
FAIR	FastAlign	$37.6_{\pm 2.2}$	$36.5_{\pm 1.1}$	$37.0_{\pm 1.5}$	$24.4_{\pm 1.2}$
FAIR	AWESOME w/o co	$66.8_{\pm 2.7}$	$63.7_{\pm 1.4}$	$65.2_{\pm 1.9}$	$49.1_{\pm 1.6}$
FAIR	AWESOME w/ co	$66.9_{\pm 2.7}$	$63.9_{\pm 1.6}$	$65.3_{\pm 1.9}$	$49.3_{\pm 1.7}$
FAIR	AWESOME ft w/o co	$68.6_{\pm 2.9}$	$63.7_{\pm 1.4}$	$66.0_{\pm 1.9}$	$49.6_{\pm 1.6}$
FAIR	AWESOME ft w/ co	$68.8_{\pm 2.8}$	$64.5_{\pm 1.4}$	$66.6_{\pm 1.9}$	$50.2_{\pm 1.6}$
FAIR	AWESOME pt+ft w/o co	$67.2_{\pm 2.8}$	$62.9_{\pm 1.4}$	$64.9_{\pm 1.9}$	$48.6_{\pm 1.6}$
FAIR	AWESOME pt+ft w/ co	$67.3_{\pm 2.9}$	$63.9_{\pm 1.6}$	$65.5_{\pm 2.0}$	$49.4_{\pm 1.7}$
FAIR ft	FastAlign	$29.1_{\pm 0.6}$	$31.1_{\pm 0.8}$	$30.1_{\pm 0.6}$	$18.3_{\pm 0.6}$
FAIR ft	AWESOME w/o co	$69.8_{\pm 2.6}$	$65.4_{\pm 1.4}$	$67.5_{\pm 1.8}$	$50.9_{\pm 3.8}$
FAIR ft	AWESOME w/ co	$68.9_{\pm 2.5}$	$66.4_{\pm 1.4}$	$67.6_{\pm 1.7}$	$51.4_{\pm 3.8}$
FAIR ft	AWESOME ft w/o co	$69.5_{\pm 2.3}$	$65.4_{\pm 1.4}$	$67.4_{\pm 1.7}$	$50.8_{\pm 3.8}$
FAIR ft	AWESOME ft w/ co	$69.7_{\pm 2.3}$	$65.5_{\pm 1.5}$	$67.5_{\pm 1.6}$	$51.0_{\pm 3.8}$
FAIR ft	AWESOME pt+ft w/o co	$\mathbf{69.9}_{\pm 2.4}$	$\mathbf{66.7}_{\pm 1.6}$	$\mathbf{68.3}_{\pm 1.8}$	$\mathbf{51.9}_{\pm 3.7}$
FAIR ft	AWESOME pt+ft w/ co	$69.2_{\pm 2.1}$	$66.2_{\pm 1.4}$	$67.6_{\pm 1.6}$	$51.3_{\pm 3.8}$
XLM-R Base					
FAIR	FastAlign	$37.0_{\pm 1.5}$	$41.2_{\pm 0.8}$	$39.0_{\pm 0.8}$	$28.4_{\pm 0.5}$
FAIR	AWESOME w/o co	$71.1_{\pm 0.7}$	$72.3_{\pm 0.8}$	$71.7_{\pm 0.3}$	$56.4_{\pm 0.6}$
FAIR	AWESOME w/ co	$71.1_{\pm 0.7}$	$72.3_{\pm 0.8}$	$71.7_{\pm 0.3}$	$56.4_{\pm 0.6}$
FAIR	AWESOME ft w/o co	$72.0_{\pm 0.7}$	$71.4_{\pm 0.8}$	$71.7_{\pm 0.3}$	$56.1_{\pm 0.7}$
FAIR	AWESOME ft w/ co	$\mathbf{72.3}_{\pm 0.7}$	$72.3_{\pm 0.8}$	$72.3_{\pm 0.3}$	$\mathbf{56.7}_{\pm 0.7}$
FAIR	AWESOME pt+ft w/o co	$70.6_{\pm 0.7}$	$70.6_{\pm 0.8}$	$70.6_{\pm 0.3}$	$55.2_{\pm 0.6}$
FAIR	AWESOME pt+ft w/ co	$70.8_{\pm 0.7}$	$71.4_{\pm 0.8}$	$71.1_{\pm 0.3}$	$55.8_{\pm 0.6}$
FAIR ft	FastAlign	$29.7_{\pm 0.7}$	$35.1_{\pm 1.1}$	$32.2_{\pm 0.7}$	$22.0_{\pm 0.8}$
FAIR ft	AWESOME w/o co	$71.0_{\pm 1.2}$	$71.9_{\pm 2.0}$	$71.4_{\pm 1.2}$	$54.3_{\pm 5.2}$
FAIR ft	AWESOME w/ co	$70.4_{\pm 0.8}$	$72.9_{\pm 1.6}$	$71.6_{\pm 0.6}$	$54.8_{\pm 5.0}$
FAIR ft	AWESOME ft w/o co	$71.7_{\pm 1.2}$	$72.4_{\pm 1.6}$	$72.1_{\pm 0.8}$	$55.1_{\pm 5.3}$
FAIR ft	AWESOME ft w/ co	$72.1_{\pm 1.2}$	$72.4_{\pm 1.6}$	$72.3_{\pm 0.7}$	$55.2_{\pm 5.2}$
FAIR ft	AWESOME pt+ft w/o co	$72.0_{\pm 1.0}$	$\mathbf{73.4}_{\pm 1.7}$	$\mathbf{72.7}_{\pm 0.8}$	$55.5_{\pm 5.0}$
FAIR ft	AWESOME pt+ft w/ co	$71.7_{\pm 1.1}$	$72.8_{\pm 2.0}$	$72.2_{\pm 1.0}$	$55.3_{\pm 5.5}$
XLM-R Large					
FAIR	FastAlign	$39.9_{\pm 1.7}$	$41.4_{\pm 0.7}$	$40.6_{\pm 1.0}$	$28.7_{\pm 1.2}$
FAIR	AWESOME w/o co	$76.4_{\pm 1.6}$	$72.5_{\pm 0.9}$	$74.4_{\pm 1.0}$	$49.2_{\pm 0.4}$
FAIR	AWESOME w/ co	$76.4_{\pm 1.6}$	$72.5_{\pm 0.9}$	$74.4_{\pm 1.0}$	$49.2_{\pm 0.4}$
FAIR	AWESOME ft w/o co	$77.5_{\pm 1.7}$	$71.6_{\pm 0.9}$	$74.5_{\pm 1.0}$	$49.0_{\pm 0.5}$
FAIR	AWESOME ft w/ co	$77.7_{\pm 1.7}$	$72.5_{\pm 0.9}$	$75.0_{\pm 1.0}$	$49.6_{\pm 0.5}$
FAIR	AWESOME pt+ft w/o co	$75.9_{\pm 1.7}$	$70.8_{\pm 0.9}$	$73.3_{\pm 1.0}$	$47.9_{\pm 0.4}$
FAIR	AWESOME pt+ft w/ co	$76.1_{\pm 1.6}$	$71.6_{\pm 0.9}$	$73.8_{\pm 1.0}$	$48.6_{\pm 0.4}$
FAIR ft	FastAlign	$30.6_{\pm 1.5}$	$34.0_{\pm 0.9}$	$32.2_{\pm 1.2}$	$20.9_{\pm 0.9}$
FAIR ft	AWESOME w/o co	$76.6_{\pm 3.5}$	$72.7_{\pm 2.1}$	$74.6_{\pm 2.7}$	$49.4_{\pm 1.8}$
FAIR ft	AWESOME w/ co	$76.5_{\pm 3.7}$	$\mathbf{74.4}_{\pm 2.1}$	$75.4_{\pm 2.8}$	$50.7_{\pm 1.9}$
FAIR ft	AWESOME ft w/o co	$77.8_{\pm 3.6}$	$73.3_{\pm 2.2}$	$75.5_{\pm 2.7}$	$50.1_{\pm 1.9}$
FAIR ft	AWESOME ft w/ co	$\mathbf{78.3}_{\pm 3.7}$	$73.3_{\pm 2.2}$	$75.7_{\pm 2.8}$	$50.2_{\pm 1.9}$
FAIR ft	AWESOME pt+ft w/o co	$77.9_{\pm 3.8}$	$\mathbf{74.4}_{\pm 2.1}$	$\mathbf{76.1}_{\pm 2.8}$	$\mathbf{51.0}_{\pm 1.9}$
FAIR ft	AWESOME pt+ft w/ co	$77.5_{\pm 3.8}$	$73.5_{\pm 2.1}$	$75.5_{\pm 2.8}$	$50.2_{\pm 1.9}$

Table 4.8 – Full results for the `translate-test` approach in German with multilingual language models.

translation	aligner	precision	recall	micro-f1	macro-f1
Opus	FastAlign	$68.5_{\pm 1.2}$	$69.1_{\pm 1.2}$	$68.8_{\pm 1.0}$	$71.2_{\pm 1.0}$
Opus	AWESOME w/o co	$75.2_{\pm 1.4}$	$\mathbf{71.8}_{\pm 1.3}$	$\mathbf{73.5}_{\pm 1.2}$	$\mathbf{75.8}_{\pm 1.2}$
Opus	AWESOME w/ co	$74.8_{\pm 1.5}$	$\mathbf{71.8}_{\pm 1.4}$	$73.3_{\pm 1.3}$	$75.6_{\pm 1.2}$
Opus	AWESOME ft w/o co	$74.7_{\pm 1.4}$	$\mathbf{71.8}_{\pm 1.3}$	$73.2_{\pm 1.2}$	$75.5_{\pm 1.2}$
Opus	AWESOME ft w/ co	$74.1_{\pm 1.4}$	$71.5_{\pm 1.3}$	$72.8_{\pm 1.2}$	$75.1_{\pm 1.2}$
Opus	AWESOME pt+ft w/o co	$74.7_{\pm 1.4}$	$\mathbf{71.8}_{\pm 1.3}$	$73.2_{\pm 1.2}$	$75.5_{\pm 1.2}$
Opus	AWESOME pt+ft w/ co	$73.3_{\pm 1.4}$	$71.5_{\pm 1.3}$	$72.4_{\pm 1.2}$	$74.9_{\pm 1.2}$
Opus ft	FastAlign	$67.9_{\pm 1.3}$	$69.3_{\pm 1.5}$	$68.6_{\pm 1.2}$	$70.6_{\pm 1.2}$
Opus ft	AWESOME w/o co	$\mathbf{75.4}_{\pm 1.3}$	$71.4_{\pm 1.6}$	$73.3_{\pm 1.3}$	$\mathbf{75.8}_{\pm 1.3}$
Opus ft	AWESOME w/ co	$74.1_{\pm 1.4}$	$71.4_{\pm 1.7}$	$72.7_{\pm 1.4}$	$75.4_{\pm 1.4}$
Opus ft	AWESOME ft w/o co	$74.7_{\pm 1.4}$	$71.4_{\pm 1.5}$	$73.0_{\pm 1.3}$	$75.4_{\pm 1.3}$
Opus ft	AWESOME ft w/ co	$74.3_{\pm 1.4}$	$71.5_{\pm 1.6}$	$72.9_{\pm 1.4}$	$75.3_{\pm 1.4}$
Opus ft	AWESOME pt+ft w/o co	$74.3_{\pm 1.3}$	$71.5_{\pm 1.6}$	$72.9_{\pm 1.3}$	$75.4_{\pm 1.3}$
Opus ft	AWESOME pt+ft w/ co	$72.3_{\pm 1.3}$	$71.5_{\pm 1.6}$	$71.9_{\pm 1.3}$	$74.7_{\pm 1.3}$

Table 4.9 – Results of `translate-test` in French with PubMedBERT.

translation	aligner	precision	recall	micro-f1	macro-f1
FAIR	FastAlign	$40.5_{\pm 2.0}$	$41.8_{\pm 1.0}$	$41.1_{\pm 1.4}$	$29.2_{\pm 1.2}$
FAIR	AWESOME w/o co	$73.6_{\pm 2.7}$	$71.9_{\pm 1.1}$	$72.7_{\pm 1.6}$	$55.1_{\pm 3.9}$
FAIR	AWESOME w/ co	$73.6_{\pm 2.7}$	$71.9_{\pm 1.1}$	$72.7_{\pm 1.6}$	$55.1_{\pm 3.9}$
FAIR	AWESOME ft w/o co	$74.7_{\pm 2.7}$	$71.1_{\pm 1.1}$	$72.8_{\pm 1.6}$	$54.9_{\pm 3.8}$
FAIR	AWESOME ft w/ co	$\mathbf{74.8}_{\pm 2.9}$	$71.9_{\pm 1.1}$	$\mathbf{73.3}_{\pm 1.7}$	$\mathbf{55.4}_{\pm 3.9}$
FAIR	AWESOME pt+ft w/o co	$73.2_{\pm 2.8}$	$70.3_{\pm 1.1}$	$71.7_{\pm 1.6}$	$53.9_{\pm 3.8}$
FAIR	AWESOME pt+ft w/ co	$73.4_{\pm 2.8}$	$71.1_{\pm 1.1}$	$72.2_{\pm 1.6}$	$54.5_{\pm 3.9}$
FAIR ft	FastAlign	$29.7_{\pm 1.8}$	$34.3_{\pm 1.7}$	$31.8_{\pm 1.6}$	$21.7_{\pm 2.6}$
FAIR ft	AWESOME w/o co	$72.6_{\pm 0.7}$	$70.8_{\pm 2.3}$	$71.6_{\pm 1.5}$	$51.9_{\pm 4.0}$
FAIR ft	AWESOME w/ co	$71.3_{\pm 0.9}$	$71.9_{\pm 2.0}$	$71.6_{\pm 1.3}$	$52.4_{\pm 3.6}$
FAIR ft	AWESOME ft w/o co	$73.4_{\pm 1.0}$	$71.1_{\pm 2.5}$	$72.2_{\pm 1.7}$	$52.7_{\pm 4.5}$
FAIR ft	AWESOME ft w/ co	$73.4_{\pm 1.0}$	$71.1_{\pm 2.5}$	$72.2_{\pm 1.7}$	$52.7_{\pm 4.5}$
FAIR ft	AWESOME pt+ft w/o co	$72.9_{\pm 0.9}$	$\mathbf{72.3}_{\pm 2.3}$	$72.6_{\pm 1.5}$	$53.1_{\pm 3.9}$
FAIR ft	AWESOME pt+ft w/ co	$72.9_{\pm 0.8}$	$71.8_{\pm 2.5}$	$72.3_{\pm 1.6}$	$53.0_{\pm 4.5}$

Table 4.10 – Results of `translate-test` in German with PubMedBERT.

model	Drug			Strength			Frequency			Duration			Dosage			Form			global	
	p	r	f1	p	r	f1	p	r	f1	p	r	f1	p	r	f1	p	r	f1	macro	micro
distilmBERT																				
CTL	52.2	82.4	63.8	**79.1**	**92.2**	**85.1**	**63.1**	**58.2**	**60.5**	87.6	88.4	88.0	67.6	67.4	67.5	55.3	38.5	44.4	68.2	65.9
+ realigned	67.8	80.9	73.8	78.5	85.9	82.0	50.3	37.9	43.2	74.7	73.5	74.1	67.9	63.2	65.4	70.9	61.3	65.8	67.4	66.4
Opus - FastAlign	**69.9**	77.0	73.2	76.3	84.3	80.1	48.7	44.7	46.6	**94.2**	**90.7**	**92.4**	64.1	70.5	67.1	67.4	62.2	64.6	**70.7**	**68.3**
Opus - AWESOME pt	68.1	82.7	74.7	77.1	89.0	82.5	40.5	39.2	39.8	93.8	**90.7**	92.2	67.0	70.8	68.8	69.7	59.8	64.2	70.4	67.9
Opus - AWESOME ft	68.0	**83.3**	**74.8**	74.4	87.1	80.1	33.7	33.2	33.4	93.8	**90.7**	92.2	66.6	70.5	68.5	68.2	60.0	63.7	68.8	66.2
Opus - AWESOME pt+fr	65.9	82.4	73.2	76.4	89.4	82.3	39.4	38.9	39.1	92.9	**90.7**	91.8	**68.4**	**72.1**	**70.2**	70.1	60.4	64.8	70.2	67.8
Opus ft - FastAlign	68.4	80.6	74.0	74.3	85.1	79.3	45.0	44.7	44.8	91.2	**90.7**	90.9	59.7	68.2	63.6	71.0	**64.3**	**67.4**	70.0	67.7
Opus ft - AWESOME pt	66.3	81.8	73.2	77.0	91.8	83.7	37.1	36.3	36.7	92.9	**90.7**	91.8	62.4	68.4	65.3	69.8	59.1	64.0	69.1	66.5
Opus ft - AWESOME ft	65.5	83.0	73.2	78.5	90.6	84.0	34.2	32.9	33.5	93.3	**90.7**	92.0	61.0	69.5	65.0	70.4	59.1	64.1	68.6	65.9
Opus ft - AWESOME pt+fr	65.0	82.7	72.8	77.6	90.2	83.4	37.3	37.9	37.6	93.3	**90.7**	92.0	59.8	69.5	64.3	**71.1**	58.5	64.1	69.0	66.2
XLM-R Base																				
CTL	83.8	83.0	83.4	87.7	97.6	92.4	**73.0**	**75.3**	**74.0**	88.8	87.9	88.3	77.4	76.6	77.0	70.9	69.5	70.2	**80.9**	**79.1**
+ realigned	**83.9**	82.8	**83.3**	87.2	98.0	92.3	66.9	69.7	68.2	90.7	90.1	90.4	71.7	67.8	69.7	71.8	66.9	69.3	78.9	76.7
Opus - FastAlign	81.2	82.7	82.0	83.2	94.9	88.6	66.9	72.1	69.4	89.6	88.4	89.0	78.5	77.6	78.1	73.9	70.3	72.0	79.8	78.2
Opus - AWESOME pt	80.5	82.4	81.4	85.3	95.3	90.0	55.3	61.6	58.1	**92.0**	**90.2**	**91.1**	78.3	**78.7**	78.4	75.2	69.7	72.2	78.6	76.4
Opus - AWESOME ft	80.5	82.4	81.4	84.9	94.9	89.6	51.0	56.1	53.4	91.5	89.8	90.6	73.1	74.2	73.6	**75.3**	69.7	72.4	76.8	74.6
Opus - AWESOME pt+fr	81.1	81.8	81.4	85.1	95.7	90.1	52.1	57.9	54.8	91.4	89.3	90.4	**79.9**	**78.7**	**79.3**	73.2	71.2	72.2	78.0	75.8
Opus ft - FastAlign	83.7	82.7	83.2	84.2	96.1	89.7	70.7	73.4	72.0	90.5	87.9	89.2	75.0	75.0	75.0	72.7	**72.3**	**72.5**	80.3	78.6
Opus ft - AWESOME pt	81.2	**83.3**	82.2	87.3	**98.4**	92.5	47.4	53.4	50.2	90.2	89.3	89.7	74.0	74.7	74.3	73.0	69.9	71.4	76.7	74.2
Opus ft - AWESOME ft	81.0	82.4	81.7	**89.0**	98.0	**93.3**	46.3	52.9	49.4	90.2	89.8	90.0	74.8	76.6	75.7	74.8	69.7	72.1	77.0	74.5
Opus ft - AWESOME pt+fr	80.7	82.4	81.5	87.0	97.3	91.9	54.9	61.6	58.0	89.8	89.3	89.5	78.9	**78.7**	78.8	72.5	69.9	71.2	78.5	76.3
XLM-R Large																				
CTL	80.3	81.5	80.9	**90.5**	97.3	**93.8**	66.0	68.9	67.3	93.2	**89.3**	91.2	76.1	75.3	75.7	75.8	67.5	71.3	80.0	77.9
+ realigned	**85.5**	82.1	83.7	90.2	97.3	93.6	**66.6**	67.6	67.1	92.7	88.8	90.7	**80.2**	77.4	**78.7**	74.2	68.0	70.9	**80.8**	**78.8**
Opus - FastAlign	82.0	80.3	81.2	87.6	96.5	91.8	65.7	69.7	**67.6**	92.3	88.4	90.3	76.5	76.1	76.3	78.7	66.2	71.9	79.8	78.0
Opus - AWESOME pt	80.8	82.7	81.7	89.8	96.1	92.8	54.8	60.0	57.3	93.2	88.8	91.0	76.7	75.5	76.0	80.3	68.0	73.5	78.7	76.5
Opus - AWESOME ft	82.7	81.2	81.9	88.1	95.7	91.7	43.9	49.5	46.5	92.7	88.4	90.5	76.8	76.8	76.8	**80.4**	67.5	73.4	76.8	74.2
Opus - AWESOME pt+fr	83.8	81.2	82.4	88.8	96.5	92.5	49.0	56.6	52.5	93.2	88.8	91.0	78.3	**77.6**	78.0	79.2	65.4	71.6	78.0	75.4
Opus ft - FastAlign	84.6	80.6	82.5	79.0	88.2	83.3	61.5	**72.1**	66.3	91.8	88.4	90.1	77.3	74.7	76.0	75.4	68.6	71.8	78.3	76.6
Opus ft - AWESOME pt	83.2	**85.1**	**84.1**	89.9	96.9	93.2	51.5	58.2	54.6	94.2	88.8	91.4	74.8	76.3	75.5	79.4	70.3	**74.6**	78.9	76.5
Opus ft - AWESOME ft	84.4	81.8	83.0	89.9	**97.6**	93.6	45.4	51.1	48.1	93.2	**89.3**	91.2	74.3	76.6	75.4	75.6	**71.4**	73.4	77.4	74.8
Opus ft - AWESOME pt+fr	82.4	80.9	81.6	89.6	**97.6**	93.4	51.4	56.1	53.6	**94.6**	**89.3**	**91.9**	74.0	76.3	75.1	75.1	70.5	72.6	78.1	75.5

Table 4.11 – Comparison class by class for `translate-train` and CTL on MedNERF with multilingual language models.

4.1 Examples from the different datasets

Examples from n2c2
The patient's agitation was managed with **nightly**$_{\text{Frequency}}$ **haldol**$_{\text{Drug}}$ with **as needed**$_{\text{Frequency}}$ **haldol**$_{\text{Drug}}$ as well.
Improvement in clinical status was noted overnight and his **morphine**$_{\text{Drug}}$ drip was discontinued.
- hold all **antihypertensives**$_{\text{Drug}}$; plan to add back slowly at reduced doses and varying schedule - rule out MI - **bolus**$_{\text{Dosage}}$ **NS**$_{\text{Drug}}$ to maintain MAP > 60 with caution given ESRD and oliguric.
folic acid$_{\text{Drug}}$ **1 mg**$_{\text{Strength}}$ **Tablet**$_{\text{Form}}$ Sig: **One (1)**$_{\text{Dosage}}$ **Tablet**$_{\text{Form}}$ PO **DAILY (Daily)**$_{\text{Frequency}}$.
Iron$_{\text{Drug}}$ **50 mg**$_{\text{Strength}}$ **Tablet**$_{\text{Form}}$ Sustained Release Sig: **One (1)**$_{\text{Dosage}}$ **Tablet Sustained Release**$_{\text{Form}}$ PO **once a day**$_{\text{Frequency}}$.
Examples from GERNERMED test set
Das **Eplerenon**$_{\text{Drug}}$ ist wegen Ihrer Herzinsuffizienz. Da können wir jetzt auf **50 mg**$_{\text{Strength}}$ p.o. 1-0-0 augmentieren. *Eplerenon is for your heart failure. We can now augment to 50mg p.o. 1-0-0.*
Wegen der COPD-Exazerbation wurde **Terbutalin**$_{\text{Drug}}$ **0,25 mg**$_{\text{Strength}}$ dem Patienten appliziert. Hierfür wurde der subkutane Weg gewählt. *Because of the COPD exacerbation, terbutaline 0.25 mg was administered to the patient. The subcutaneous route was chosen for this.*
Bei Vorhofflimmern ist neben **Betablockern**$_{\text{Drug}}$ auch die Gabe von **Magnesium**$_{\text{Drug}}$ p.o. sinnvoll. Hierfür würden wir mit **300 mg**$_{\text{Strength}}$ **einmal täglich**$_{\text{Frequency}}$ starten. Sofern möglich, ist eine Einnahme **mittags**$_{\text{Frequency}}$ (ca. 12 Uhr) zu bevorzugen. *In atrial fibrillation, in addition to beta-blocks, the administration of magnesium p.o. is also useful. For this we would start with 300 mg once a day. If possible, it is preferable to take it at noon (around 12 o'clock).*
Zur Optimierung der Herzinsuffizienztherapie wurde die Dosis von **Sacubitril / Valsartan**$_{\text{Drug}}$ auf **97 / 103 mg**$_{\text{Strength}}$ in **Tablettenform**$_{\text{Form}}$ mit Einnahme **am Morgen und am Abend**$_{\text{Frequency}}$ erweitert. *To optimize heart failure therapy, the dose of sacubitril / valsartan was extended to 97 / 103 mg in tablet form with intake in the morning and evening.*
Bei bekannter koronarer Herzerkrankung sollte **lebenslang**$_{\text{Duration}}$ **Acetylsalicylsäure**$_{\text{Drug}}$ **100 mg**$_{\text{Strength}}$ **morgens täglich**$_{\text{Frequency}}$ in oraler Applikation eingenommen werden. *In cases of known coronary artery disease, acetylsalicylic acid 100mg should be taken orally daily in the morning as a lifelong treatment.*
Examples from MedNERF
TRAMADOL / PARACETAMOL$_{\text{Drug}}$ **37,5mg / 325mg**$_{\text{Strength}}$ *TRAMADO/PARACETAMOL 37,5mg/325mg*
AMLODIPINE$_{\text{Drug}}$ **5 mg**$_{\text{Strength}}$; **cpr**$_{\text{Form}}$ **1**$_{\text{Dosage}}$ **comprimé**$_{\text{Form}}$ **matin**$_{\text{Frequency}}$ **1**$_{\text{Dosage}}$ **comprimé**$_{\text{Form}}$ **soir**$_{\text{Frequency}}$ *AMLODIPINE 5mg; tab 1 tablet in the mording 1 tablet in the evening*
DOLIPRANETABS$_{\text{Drug}}$ **1000 MG**$_{\text{Strength}}$ **CPR PELL**$_{\text{Form}}$ PLQ / 8 (**Paracétamol**$_{\text{Drug}}$ **1.000 mg**$_{\text{Strength}}$ **comprimé**$_{\text{Form}}$) *DOLIPRANETABS 1000mg TAB PLQ / 8 (Paracetamol 1,000mg tablet)*
ACIDE ACETYLSALICYLIQUE$_{\text{Drug}}$ (**sel de lysine**$_{\text{Drug}}$) **75 mg**$_{\text{Strength}}$ **pdre p sol buv sach**$_{\text{Form}}$ (**KARDEGIC**$_{\text{Drug}}$) *ACETYLSALICYLIC ACID (lysine salts) 75mg oral powder for suspension (KARDEGIC)*
1$_{\text{Dosage}}$ **sachet**$_{\text{Form}}$ **matin midi et soir**$_{\text{Frequency}}$ si besoin *1 packet in the morning, at noon, and in the evening, if needed*

Chapter 5

Résumé en français: Comprendre et évaluer les embeddings multilingues non supervisés dans les domaines général et clinique

Cette thèse vise à améliorer la compréhension des embeddings[23] multilingues et est motivée par des applications dans le domaine clinique. Elle comble certaines lacunes dans la littérature sur les embeddings multilingues en général, puis évalue si les embeddings multilingues appris dans le domaine général peuvent être utilisés dans le domaine clinique, notamment pour une tâche de reconnaissance d'entités nommées (NER).

Différents types de modèles linguistiques peuvent produire différents embeddings. Cette thèse se concentre sur deux types principaux :

- Les word embeddings statiques (SWEs), pour lesquels chaque mot se voit attribuer un embedding fixe ;
- Les word embedings contextualisés (CWEs), où l'embedding d'un mot varie en fonction du contexte dans lequel il se trouve.

Les CWEs ont l'avantage de permettre différentes représentations pour différents sens d'un mot polysémique. Par exemple, le mot "tie" peut avoir un embedding différent lorsqu'il est trouvé dans la phrase "She's wearing a tie" ou dans la phrase "The match ended with a tie". Cependant, les SWEs ont également leurs avantages. L'un d'eux est qu'une fois qu'un embedding est entraîné, la représentation d'un mot donné peut être obtenue par une simple table de recherche.

Les SWEs sont principalement construits avec l'idée que les mots de sens similaire doivent avoir des représentations proches. Cependant, la similarité distributionnelle qui découle du partage de contextes similaires peut encapsuler une grande variété de relations : synonymie, antonymie, mais aussi des valeurs sur une échelle (comme les chiffres ou les noms de mois), l'hyponymie et l'hypernymie (comme "cheval" et "animal") (Turney and Pantel, 2010). Il existe également des relations plus complexes qui pourraient être impliquées, comme la similarité attributionnelle, où les mots de concepts partageant des caractéristiques saillantes peuvent avoir une similarité distributionnelle élevée, comme "chaise" et "cheval", qui ont tous deux quatre pattes et un dos.

Alors que les CWEs peuvent être vus comme des SWEs améliorées capables de gérer la polysémie en tenant compte du contexte, ils ne sont pas toujours construits dans la même but. Les représentations statiques sont explicitement censées refléter une certaine similarité sémantique, et différents SWEs sont souvent comparés en utilisant la corrélation avec le jugement humain de la similarité sémantique.

[23] aussi appelés plongements lexicaux

En revanche, les CWEs sont plus souvent comparés sur un ensemble de tâches finales (Devlin et al., 2019). Ils sont généralement évaluée sur chaque tâche en entraînant davantage le modèle sur la tâche donnée et en l'évaluant sur un ensemble de test indépendant. Cette entraînement spécifique à la tâche, qui s'ajoute au pré-entraînement non supervisé du modèle, est appelée fine-tuning, et le processus selon lequel on espère que le fine-tuning fournira de meilleurs résultats que l'entraînement d'un modèle à partir de zéro est appelé apprentissage par transfert.

Comment les embeddings statiques et contextualisés peuvent-ils être étendus à un contexte multilingue ?

Il existe deux principaux types de capacités multilingues largement étudiées dans la littérature :

- L'alignement multilingue (MA)
- L'Apprentissage par Transfert translingue (CTL)

Avec l'alignement multilingue (MA), l'objectif est de construire un Word Embedding (WE) qui soit cohérent entre les langues. Par conséquent, si un embedding monolingue doit représenter des mots de sens similaires avec des représentations proches, alors un embedding multilingue aligné doit avoir la même exigence fonctionnant à travers les langues, ce qui signifierait en pratique que chaque mot doit être proche de sa traduction.

En revanche, avec l'Apprentissage par Transfert translingue (CTL), l'objectif est de construire un modèle de langage multilingue (mLM) qui, lorsqu'il est entraîné sur une tâche secondaire dans une langue, peut ensuite être en mesure d'effectuer cette tâche de manière précise dans une autre langue.

Les embeddings statiques multilingues sont souvent construits en tenant compte de l'alignement multilingue, tandis que les embeddings contextualisés peuvent montrer de fortes capacités de CTL, souvent sans aucun signal d'entraînement multilingue (Devlin et al., 2019). Cette thèse passe en revue d'abord la littérature sur les embeddings de mots multilingues, puis aborde certaines lacunes dans la littérature concernant le lien entre l'alignement multilingue et l'CTL, et montre enfin que les embeddings contextualisés multilingues de domaine général peuvent être utilisés dans le domaine clinique, et correspondent aux performances de méthods compétitives basées sur la traduction.

5.1 Contexte

Cette section propose une typologie des modèles de langage multilingues existants en fonction de deux caractéristiques principales :

- Le niveau de supervision translingue (*cross-lingual*) : le modèle repose-t-il sur des données parallèles ? avec quelques nuances supplémentaires concernant le type de signal d'entraînement.
- Le type de modèle de langage ou de représentation linguistique, ou plus précisément le type de donnée qu'ils produisent, qu'il s'agisse d'un embedding (statique ou contextualisé) ou d'un modèle de langage génératif produisant du texte.

5.1.1 Différents niveaux de supervision translingue

Les modèles de langage monolingues sont généralement entraînés avec un objectif d'entraînement non supervisé - ou plutôt auto-supervisé - comme la prédiction de mots masqués (MLM) (Devlin et al., 2019) pour les embeddings contextualisés ou skip-gram (Mikolov et al., 2013) pour les embeddings statiques. Pour les MLMs multilingues, les mêmes objectifs monolingues peuvent être utilisés et un objectif translingue peut être ajouté, qui peut être schématisé comme suit :

$$J = \mathcal{L} + \Omega$$

$\mathcal{L}$ représente l'objectif de pré-entraînement monolingue, qui peut généralement être considéré comme une somme d'objectifs monolingues pour chaque langue impliquée. Ω est l'objectif de

pré-entraînement translingue, qui vise à fournir un certain niveau d'alignement multilingue. Ce terme de pré-entraînement translingue peut être supervisé ou non supervisé.

Si l'objectif translingue est non supervisé, deux cas peuvent être différenciés :

- langage-dépendant : lorsque le terme translingue est explicite, comme dans les méthodes basées sur un alignement non supervisée pour les embeddings statiques de mots multilingues.
- langage-indépendant : lorsqu'aucun signal translingue explicite n'est utilisé. mBERT (Devlin et al., 2019) ou XLM-R (Conneau et al., 2020b) sont des exemples de modèles de langage langage-indépendants.

Un modèle sera langage-dépendant dès lors qu'il aura une composante spécifique à une langue. Par exemple, MAD-X (Pfeiffer et al., 2020) rajoute des paramétres spécifiques à chaque langue dans un modèle existant, il est donc considéré comme langage-dépendant. Des modèles comme mBART (Liu et al., 2020b) utilisent un symbole spécial précisant la langue donnée en entrée et peuvent donc également être considérés comme langage-dépendants.

Si l'objectif translingue est supervisé, on peut définir deux niveaux différents de supervision :

- Supervision au niveau de la phrase : où seul un corpus parallèle est nécessaire, et où la similarité entre les représentations de phrases est explicitement imposée.
- Supervision au niveau du mot : où le signal d'entraînement consiste en paires de mots traduits.

Pour les embeddings statiques, les paires de mots peuvent être plus difficiles à obtenir que les paires de phrases. Cependant, le signal de supervision de l'alignement au niveau des mots peut être encore plus rare pour les embeddings contextualisés, où les paires de mots doivent être replacées dans un contexte, c'est-à-dire que les paires de mots doivent être extraites à partir de paires de phrases elles-mêmes traduites.

5.1.2 Différents types de modèles de langage et de représentations

Différents types de modèles de langage ont été appliqués dans un contexte multilingue. Cette étude se concentre sur les embeddings de mots produits par des réseaux neuronaux peu profonds tels que les embeddings statiques ou par des Transformers à encodeur seul pour les embeddings contextualisés. Cependant, la typologie proposée doit également inclure des modèles de langage génératifs.

5.1.3 La typologie proposée

La typologie proposée est présentée dans le Tableau 5.1 avec quelques exemples et révèle certaines frontières de l'état de l'art.

La première concerne les embeddings statiques : il n'existe pas d'embedding statique langage-indépendant. Étant donné que les méthodes non supervisées langage-dépendants sont difficiles à mettre en pratique dans de nombreuses situations (Søgaard, Ruder, and Vulić, 2018), on pourrait anticiper que toute tentative d'embedding statique langage-indépendant ne fournirait pas d'embedding bien aligné. Cependant, cette thèse tient à revisiter les embeddings statiques multilingues avec la nouvelle perspective langage-indépendante apportée par des modèles plus grands. Elle ne fournit pas de nouveaux embeddings état-de-l'art avec une grande précision en Induction de Dictionaire Bilingue (BLI), mais elle apporte un autre éclairage sur l'échec des méthodes d'alignement non supervisées existantes et sur le lien entre l'alignement et le pré-entraînement.

Alors que les embeddings statiques langage-indépendants sont une voie de recherche ouverte, il en existe une autre à l'autre extrémité des embeddings multilingues : la supervision translingue au niveau des mots pour les embeddings contextualisés. Bien que des méthodes de réalignement au niveau des mots existent, elles ne sont pas encore systématiquement efficaces (Wu and Dredze, 2020). Cette thèse examine l'échec de ces méthodes de réalignement et étudie le lien entre l'alignement et le CTL.

	embeddings		Modèles génératifs	
	statiques	contextualisés (encodeur seul)	texte-texte (encodeur-décodeur)	auto-régressifs (décodeur seul)
Non supervisé				
langage-indépendan		- Transformers à encodeur seul (mBERT, XLM-R)	- encodeur-décodeur entraîné avec un objectif de débruitage (mT5)	- décodeur seul entraîné sur la génération de texte (XGLM, Bloom) - post-entraînement avec des tâches spécifiques (BloomZ, mT0)
langage-dépendant	- méthodes basées sur l'alignement - adversariales, ICP (MUSE) - correspondance de graphes (RCSLS, VecMap) - alignement joint (IsoVec)	- dé-identification de la langue (lng-free) - adaptateurs (MAD-X)	- mot spécial pour chaque langue (mBART) - adaptateurs (mmT5)	
Supervisé				
Phrases	méthodes d'alignement à l'échelle des phrases[1]	- TLM - réalignement à l'échelle des phrases	- modèles de traduction	
Mots	- basé sur l'alignement (Procrustes) - pseudo-mixing - alignement joint	- réalignement		

Table 5.1 – Typologie proposée des modèles multilingues avec des exemples. Les cellules **orange** montrent les frontières de la littérature explorées par cette thèse.

embedding	BLI normal	BLI facile
embeddings non alignés	0,00%	0,00%
langage-indépendant	0,00%	34,53%
supervisé (Procrustes)	43,3%	74,7%

Table 5.2 – Résultats de BLI pour un embedding de mots bilingues langage-indépendant par rapport à une référence supervisée d'alignement d'embeddings de mots. "BLI normale" est le résultat de la BLI lorsque l'espace de recherche est l'ensemble du vocabulaire cible, tandis que "BLI facile" est la précision lorsque l'espace de recherche est réduit au vocabulaire du dictionnaire bilingue.

5.2 embeddings statiques langage-indépendants

Cette section se penche sur la question de savoir si des représentations statiques multilingues peuvent être obtenues avec un pré-entraînement langage-indépendant de la même manière que les embeddings contextualisés.

À cette fin, FastText (Bojanowski et al., 2017), un algorithme populaire pour l'apprentissage de embeddings statiques, est entraîné sur un corpus construit en concaténant Wikipédia en anglais et en chinois. Le corpus chinois est répété de manière à constituer près de la moitié des données de pré-entraînement.

Comme le montre le Tableau 5.2, lorsqu'il est évalué en utilisant un dictionnaire de test populaire de 1 500 paires provenant de Lample et al. (2018), il fournit une précision de 0,0% en top-1, alors qu' un alignement supervisé d'embeddings monolingues peut atteindre plus de 40% pour la paire anglais-chinois. À première vue, un entraînement langage-indépendant ne semble donc pas fournir de capacités multilingues à un embedding statique, contrairement aux embeddings contextualisés.

Cependant, l'embedding statique langage-indépendant est en réalité légèrement aligné. Si l'espace de recherche pour la recherche du plus proche voisin est réduit, la précision de la BLI augmente. Par exemple, si nous réduisons l'espace de recherche aux mots présents uniquement dans les 1 500 paires du dictionnaire bilingue, la précision de BLI augmente à 34,53%. Mais il reste néanmoins en deçà du résultat compétitif de la référence supervisée.

Mais qu'est-ce qui, dans les données d'entraînement, peut provoquer ce léger alignement ? Une

corpus d'entraînement	BLI facile
`originale`	34,53%
`no-code-switch`	2,44%
`no-special`	38,07%
`script-only`	0,00%

Table 5.3 – Résultats de l'analyse d'ablation avec BLI "facile" (espace de recherche réduit).

méthode	zh	ar	af	bs	th	tl	ta
Méthodes basées sur l'alignement non supervisé							
MUSE	$0{,}0_{\pm 3{,}3}$	$30{,}9_{\pm 3{,}3}$	$4{,}5_{\pm 8{,}9}$	$0{,}0_{\pm 0{,}0}$	$0{,}0_{\pm 0{,}0}$	-*	$0{,}0_{\pm 0{,}0}$
WP	$0{,}6_{\pm 0{,}8}$	$10{,}7_{\pm 9{,}9}$	$23{,}5_{\pm 1{,}1}$	$0{,}5_{\pm 0{,}6}$	$0{,}0_{\pm 0{,}0}$	$3{,}0_{\pm 3{,}0}$	$0{,}0_{\pm 0{,}0}$
VecMap	$0{,}0_{\pm 0{,}0}$	$36{,}1_{\pm 1{,}8}$	$33{,}7_{\pm 1{,}0}$	$0{,}1_{\pm 0{,}2}$	$0{,}0_{\pm 0{,}0}$	$0{,}1_{\pm 0{,}1}$	$0{,}1_{\pm 0{,}1}$
Vecmap avec différentes initialisations							
Vecmap avec MUSE	$6{,}5_{\pm 12{,}6}$	$\underline{39{,}5}_{\pm 0{,}7}$	$7{,}2_{\pm 14{,}0}$	$0{,}1_{\pm 0{,}2}$	$0{,}0_{\pm 0{,}0}$	-*	$3{,}0_{\pm 5{,}9}$
Vecmap avec WP	$0{,}0_{\pm 0{,}1}$	$31{,}9_{\pm 15{,}9}$	$\underline{34{,}7}_{\pm 0{,}4}$	$0{,}2_{\pm 0{,}1}$	$0{,}0_{\pm 0{,}0}$	$8{,}7_{\pm 6{,}8}$	$0{,}0_{\pm 0{,}0}$
Vecmap identique	$\mathbf{36{,}8}_{\pm 0{,}8}$	$\mathbf{39{,}8}_{\pm 0{,}3}$	$\mathbf{\underline{34{,}9}}_{\pm 0{,}4}$	$\mathbf{28{,}3}_{\pm 0{,}8}$	$\mathbf{24{,}0}_{\pm 0{,}5}$	$\mathbf{19{,}4}_{\pm 0{,}4}$	$\mathbf{17{,}8}_{\pm 0{,}3}$
Procrustes supervisé avec ou sans auto-apprentissage							
supervisé avec auto-apprentissage	$38{,}1_{\pm 0{,}4}$	$39{,}5_{\pm 0{,}1}$	$34{,}5_{\pm 0{,}4}$	$28{,}5_{\pm 0{,}3}$	$24{,}5_{\pm 0{,}2}$	$19{,}1_{\pm 0{,}6}$	$18{,}3_{\pm 0{,}5}$
supervisé	43,3	43,0	34,8	28,9	25,1	22,8	20,2

* MUSE n'était pas applicable pour cette langue avec les paramètres par défaut car il nécessite un vocabulaire d'au moins 75 000 mots et l'embedding Wikipedia pour cette langue en a moins.

Table 5.4 – Résultats de différentes méthodes pour l'alignement de l'anglais vers certaines langues éloignées. La méthode d'évaluation est le Induction de Dictionaire Bilingue (BLI) avec des dictionnaires fournis par Lample et al. (2018). La précision rapportée est la précision top-1 pour le critère CSLS (Joulin et al., 2018) pour la traduction de l'anglais vers la langue cible. Les langues sont triées par ordre décroissant des résultats de la méthode supervisée. À l'exception de "supervisé", qui est déterministe, toutes les méthodes ont été appliquées 5 fois avec différentes initialisation et l'écart-type est rapporté. Le gras indique la meilleure méthode non supervisée pour une langue donnée et les valeurs qui se trouvent dans son écart-type sont soulignées.

analyse d'ablation sur les données de pré-entraînement est effectuée pour en identifier la raison. L'anglais et le chinois ont été choisis pour des raisons pratiques car ils ne s'appuient pas sur le même script. Quatre embeddings sont obtenus avec quatre données prétraitées différentes. Dans la version `originale`, les corpus monolingues sont conservés tels quels. Dans `no-code-switch`, le code-switching est supprimé grâce aux différents systèmes d'écriture. En revanche, `no-special` supprime tous les mots potentiellement communs (ponctuation, chiffre...). Enfin, `script-only` supprime à la fois le code-switching et les mots communs.

Le Tableau 5.3 montre les résultats de l'analyse d'ablation. Il démontre que lorsque le code-switching est supprimé (`no-code-switch`), l'alignement est réduit. En revanche, la suppression de la ponctuation et des chiffres ne réduit pas l'alignement, mais l'augmente même (`no-special`). Cependant, c'est sans les deux (`script-only`), lorsque les vocabulaires des deux langues n'ont vraiment aucune intersection, que la précision de l'alignement tombe à zéro.

Cette section montre que les embeddings statiques langage-indépendants sont possibles, mais qu'ils ne sont pas aussi performants que leurs homologues contextualisés. Le code-switching est également démontré comme un signal translingue utile qui se produit naturellement, et certaines expériences supplémentaires, qui ne sont pas reproduites dans ce résumé, montrent que ce signal peut être exploité pour produire des embeddings statiques multilingues non supervisés compétitifs.

5.3 Limitations des embeddings statiques multilingues

Søgaard, Ruder, and Vulić (2018) ont montré que les méthodes d'alignement non supervisées basées sur l'alignment non supervisé d'embeddings statiques multilingues présentent certaines limitations. En particulier, il est difficile d'apprendre un alignement entre les embeddings de langues éloignées,

méthode	fr	it	ru	hu
MUSE	$0{,}0_{\pm 0{,}0}$	$0{,}0_{\pm 0{,}0}$	$0{,}0_{\pm 0{,}0}$	$0{,}0_{\pm 0{,}0}$
WP	$0{,}0_{\pm 0{,}1}$	$0{,}0_{\pm 0{,}0}$	$0{,}0_{\pm 0{,}0}$	$0{,}0_{\pm 0{,}0}$
VecMap	$0{,}0_{\pm 0{,}0}$	$0{,}0_{\pm 0{,}0}$	$0{,}0_{\pm 0{,}0}$	$0{,}0_{\pm 0{,}0}$
Vecmap identique	$20{,}1_{\pm 0{,}4}$	$12{,}0_{\pm 0{,}0}$	$1{,}5_{\pm 0{,}1}$	$1{,}1_{\pm 0{,}1}$
supervisé avec auto-apprentissage	$20{,}2_{\pm 0{,}6}$	$16{,}7_{\pm 0{,}2}$	$3{,}6_{\pm 1{,}0}$	$4{,}5_{\pm 0{,}4}$
supervisé	42,7	36,1	25,1	23,2

Table 5.5 – Résultats de quelques méthodes avec des embeddings de domaines différents.

et il en va de même pour les embeddings de domaines différents.

Cette section examine ces deux principales limitations et détermine si le manque d'isométrie, l'instabilité de l'initialisation dans les méthodes basées sur l'alignement ou l'inadéquation de l'auto-apprentissage sont à l'origine de cette limitation. La plupart des méthodes d'alignement non supervisées peuvent être décomposées en deux étapes : une initialisation, où un premier alignement ou un dictionnaire est trouvé, et l'auto-apprentissage, où l'alignement est affinée de manière itérative.

Plusieurs méthodes basées sur l'alignement sont comparées sur plusieurs langues et selon deux paramétrages : lorsque les embeddings proviennent du même domaine (Wikipedia) ou de domaines différents (Wikipedia et PubMed).

Les résultats sont présentés dans le Tableau 5.4. Ils montrent qu'il peut y avoir un manque d'isométrie entre certaines langues, car même la méthode supervisée peut avoir de faibles résultats pour des langues comme le tamoul (ta) ou le thaï (th).

Mais un autre facteur qui pourrait entraver les méthodes basées sur l'alignement non supervisé est la qualité de l'initialisation. En effet, certaines initialisations comme "Vecmap identique" (Søgaard, Ruder, and Vulić, 2018) produisent des résultats proches des méthodes supervisées, tandis que des initialisations comme l'apprentissage adversaire de MUSE semblent fournir des résultats moins stables, car elles sont surpassées par la méthode Vecmap originale sur certaines langues et présentent une plus grande écart type.

Enfin, et dans une moindre mesure, la procédure d'auto-apprentissage a également son importance. La procédure d'auto-apprentissage de Vecmap (supervisé avec auto-apprentissage) ne dégrade pas trop les résultats par rapport à la méthode supervisée sans auto-apprentissage. Cependant, lorsque l'on examine MUSE ou WP, la procédure d'auto-apprentissage de Vecmap semble fournir de meilleurs résultats que l'original : Vecmap avec MUSE est meilleur que MUSE et Vecmap avec WP est meilleur que WP.

Cette thèse effectue également une expérience similaire, mais dans un contexte où les embeddings des deux langues sont obtenus à partir de corpus de domaines différents. Le Tableau 5.5 montre la précision de l'alignement entre un embedding en anglais construit à partir du corpus PubMed[24] et un embedding pour différentes langues cibles construit à partir de Wikipedia.

Les résultats montrent que le problème est très différent dans un contexte de domaines différents par rapport aux langues éloignées. Alors que l'approche supervisée sans auto-apprentissage peut encore obtenir des résultats relativement bons, les autres méthodes ont du mal à atteindre même la moitié de la précision de la méthode la plus compétitive. Dans le contexte de domaines différents, il semble y avoir un problème de stabilité de la procédure d'auto-apprentissage, car les deux méthodes supervisées obtiennent des résultats très différents avec et sans auto-apprentissage.

D'autres expériences de cette thèse étudient le manque apparent d'isométrie entre des domaines différents et montrent que ce contexte présente une importante limitation spécifique : un écart entre les vocabulaires. La procédure d'auto-apprentissage des méthodes d'alignement existantes tente généralement d'aligner l'ensemble du embedding en même temps, ou plus précisément un grand ensemble des mots les plus fréquents de chaque embedding, généralement les 20 000 mots les plus fréquents dans Vecmap. Cependant, entre deux domaines différents, la distribution du vocabulaire peut varier. Certains mots peuvent être plus fréquents dans un domaine et moins dans l'autre, voire

[24] Un corpus d'articles scientifiques biomédicaux.

absents.

Par conséquent, d'autres expériences sont rapportées dans la thèse mais pas dans ce résumé, montrant que lorsque l'alignement entre des embeddings de domaines différents a une certaine précision non nulle, les deux ensembles de embeddings ne sont en réalité qu'en partie alignés, probablement en fonction du vocabulaire commun.

5.4 Comparaison de l'alignement de embeddings statiques et contextuels

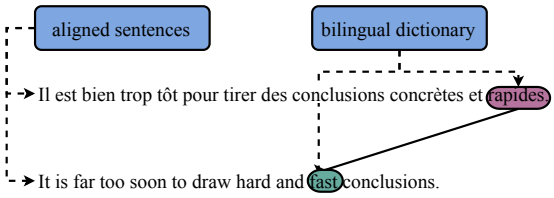


Figure 5.1 – Extraction d'une paire traduite contextualisée avec un dictionnaire bilingue.

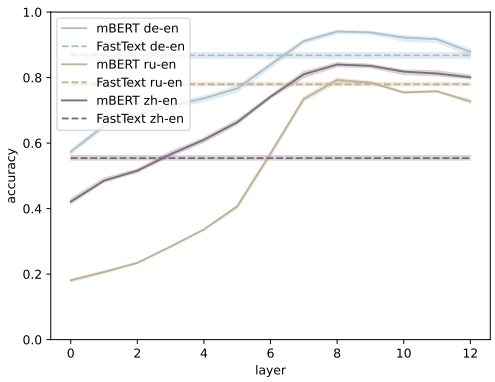


Figure 5.2 – Évolution de l'alignement à travers les couches pour mBERT comparé à FastText aligné.

Les sections précédentes ont montré que les méthodes non supervisées existantes pour construire des embeddings statiques multilingues présentent des limitations connues et ne fournissent pas toujours un bon alignement multilingue. En revanche, la littérature existante a montré que les embeddings contextuels comme mBERT peuvent offrir de bonnes propriétés multilingues sans signal de supervision explicite pour le multilinguisme. Mais ces embeddings contextuels sont généralement évalués pour leurs capacités de transfert translingue plutôt que pour leur alignement multilingue. Et il n'y a pas de véritable consensus dans la littérature sur la question de savoir si les embeddings contextuels sont effectivement bien alignés.

Ce chapitre propose une méthode d'évaluation de l'alignement multilingue dans les embeddings contextuels. Basée sur une recherche des plus proches voisins, comme Induction de Dictionnaire Bilingue (BLI), elle permet une comparaison directe avec les embeddings statiques.

La méthode d'évaluation proposée nécessite deux ensembles de données : un ensemble de données de traduction contenant des paires de phrases traduites et un dictionnaire bilingue. Le dictionnaire bilingue contient des paires de mots traduits, comme la paire "rapide"-"fast" dans notre exemple Fig. 5.1. Dans une paire de phrases traduites de l'ensemble de données de traduction, pour chaque mot d'une phrase, tous les mots traduits potentiels indiqués par le dictionnaire bilingue sont collectés à partir de l'autre phrase. Une paire de mots traduits est conservée avec son contexte s'il n'y a qu'un seul candidat pour la traduction.

La tâche de recherche est similaire à l'évaluation de BLI pour les embeddings statiques. Le score d'évaluation est la précision top-1 de la recherche de la traduction parmi les représentations d'un ensemble donné de paires de mots traduits contextuels.

Les résultats pour mBERT et cinq paires de langues sont présentés sur la Figure 5.2. Elle montre que pour quelques couches profondes, mBERT produit des représentations plus fortement alignées que les embeddings statiques multilingues.

Des expériences supplémentaires sont menées dans la thèse. D'autres modèles sont comparés, montrant que mBERT a l'un des meilleurs alignements multilingues. Différentes représentations au niveau de la phrase sont également évaluées, montrant que l'utilisation du premier mot ou du symbole spécial "[CLS]" pour constuire une représentation de la phrase n'est pas adaptée dans un cadre multilingue. Il est également démontré que l'utilisation d'un dictionnaire bilingue pour

extraire des paires de mots traduits contextuels est préférable par rapport aux outils d'alignement de mots comme FastAlign. Enfin, lorsqu'il est évalué sur une forme plus stricte d'alignement, appelée alignement fort (Roy et al., 2020), les embeddings contextuels comme mBERT surpassent largement une méthode basée sur des embeddings statiques explicitement alignés.

5.5 Le lien entre l'alignement multilingue et le transfert translingue

Cette section étudie le lien entre l'alignement et le transfert translingue pour des embeddings contextuels multilingues tels que mBERT et XLM-R. Elle mesure l'alignement au niveau des mots de la même manière que la section précédente et le compare avec la capacité de transfert translingue pour différentes langues, modèles et initialisation aléatoires.

tâche	corrélation
POS	0,87
NER	0,70
NLI	0,82

Table 5.6 – Corrélation (Spearman) entre CTL et alignement de la dernière couche après fine-tuning, pour 5 langues, 5 initialisations aléatoires et 4 modèles ($N = 100$, $p < 0,05$).

Un modèle a de bonnes capacités de transfert translingue lorsque après un ajustement fin pour une langue, il peut obtenir un bon score d'évaluation sur d'autres langues. Pour l'évaluer pour une tâche donnée, cette thèse propose de calculer la différence relative entre la métrique d'évaluation m_{en} sur l'ensemble de développement en anglais et la métrique d'évaluation m_{tgt} sur la langue cible :

$$\text{transfert translingue} = \frac{m_{\text{tgt}} - m_{\text{en}}}{m_{\text{en}}} \tag{5.1}$$

Ce score de CTL est comparé aux métriques d'alignement définies dans la section précédente. Les résultats montrent qu'il y a une forte corrélation entre les deux, et ce, pour trois tâches différentes : POS-tagging, NER et inférence de langage (NLI).

Un résumé des résultats est présenté dans le Tableau 5.6, bien que la thèse contienne des résultats plus détaillés avec des comparaisons entre différentes couches, avant et après l'ajustement fin sur une tâche spécifique, et avec différents types d'alignement.

Cette thèse constate une forte corrélation entre le transfert translingue et l'alignement multilingue. Cependant, la littérature montre également, et cela est confirmé par d'autres expériences rapportées dans la thèse, que l'amélioration de l'alignement de tels modèles n'entraîne pas systématiquement de meilleures capacités de généralisation translingue. D'autres expériences identifient deux causes principales de l'échec des méthodes de réalignement.

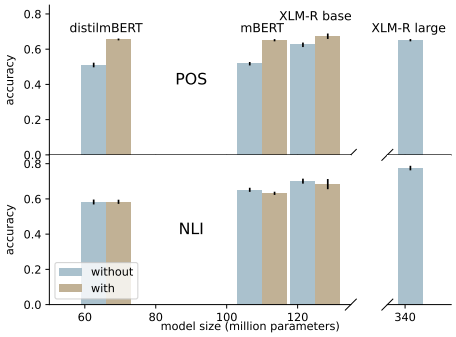


Figure 5.3 – Transfert translingue entre l'anglais et l'arabe avec et sans réalignement, en utilisant un dictionnaire bilingue. Pour certaines tâches, le réalignement peut rendre les petits modèles compétitifs avec une ligne de base plus grande.

La première cause d'échec du réalignement est l'utilisation d'outils d'alignement de mots sujets à des erreurs comme FastAlign. Cette section résout ce problème en proposant la même extraction basée sur des dictionnaires bilingues que celle proposée dans la section précédente, et montre des améliorations pour le étiquetage grammatical (POS-tagging).

La deuxième cause d'échec est lorsque le modèle voit déjà son alignement multilingue s'améliorer lorsqu'il est affiné sur la tâche de fine-tuning. Il semblerait que le modèle peut parfois être déjà suffisamment bien aligné pour la tâche en question, de telle sorte que tout réalignement supplémentaire n'améliore pas significativement les performances. Pour vérifier cette hypothèse, la Figure 5.3 compare le transfert translingue entre l'anglais et l'arabe pour plusieurs tâches, avec ou sans réalignement, en utilisant un dictionnaire bilingue.

Néanmoins, cette section constate que le réalignement fonctionne bien pour les tâches de bas niveau et les petits modèles multilingues, permettant ainsi à des modèles plus petits comme

distilmBERT (66 millions de paramètres) d'être compétitifs avec XLM-R Large (354 millions de paramètres) pour du POS-tagging, comme illustré dans la Figure 5.3.

5.6 Applications au NER dans le domaine clinique

Cette section compare les modèles contextualisés à des méthodes basées sur la traduction. Alors que cela avait déjà été fait dans le domaine général (Yarmohammadi et al., 2021), on se concentre ici sur le domaine clinique, où des modèles de langage spécifiques au domaine et des modèles de traduction peuvent être utilisés.

Cette thèse propose MedNERF[25]. Il s'agit d'un ensemble de données d'extraction d'entités nommées médicales en français. Il a été construit à partir d'un échantillon d'ordonnances médicales en français annotées selon les mêmes consignes que l'ensemble de données n2c2 (Henry et al., 2019). n2c2 fournit ainsi un ensemble d'entraînement et MedNERF un ensemble de test pour la généralisation translingue, et ils sont utilisés pour une comparaison contrôlée du transfert translingue avec des modèles multilingues de domaine général [26], avec des méthodes basées sur la traduction, qui peuvent exploiter des modèles de traduction spécifiques au domaine pour s'appliquer soit à l'ensemble d'entraînement, soit à l'ensemble de test, et qui permettent également de s'appuyer sur des modèles de langage monolingues spécifiques au domaine pour effectuer les prédictions.

Les résultats de la thèse montrent que le transfert translingue avec des modèles multilingues est efficace pour une tâche spécifique au domaine comme le NER clinique, donnant des résultats comparables à la traduction de l'ensemble d'entraînement. Cependant, le CTL a l'avantage de fonctionner immédiatement, tandis que les méthodes basées sur la traduction nécessitent de choisir soigneusement les modèles de traduction et d'alignement. Le choix de ces modèles basé sur des valeurs intrinsèques spécifiques au domaine, comme les scores de réglage fin sur des données parallèles cliniques, ou l'utilisation d'un modèle de langage spécifique au domaine, ne fournit pas de résultats sensiblement meilleurs pour la tâche finale dans la langue cible. Il est montré que la sélection du modèle d'alignement est particulièrement cruciale, et les résultats des méthodes basées sur la traduction pourraient probablement être améliorés par un post-traitement de l'alignement. De plus, le CTL offre également une marge de progression, car le réalignement des représentations des modèles peut augmenter considérablement les résultats dans certains cas.

Le pré-entraînement d'un modèle multilingue avec des données cliniques est une piste intéressante pour de futures améliorations dans le domaine de la NER clinique multilingue. Bien que les résultats montrent que l'utilisation d'un modèle monolingue spécifique au domaine avec la traduction n'est pas à la hauteur des modèles multilingues à usage général, ils montrent également que le modèle clinique en français DrBERT offre les meilleurs résultats pour les méthodes basées sur la traduction lorsqu'il utilise le modèle biomédical en anglais PubmedBERT comme initialisation. Ce modèle "presque bilingue" et spécifique au domaine permet déjà d'avoir de meilleurs résultats qu'avec un modèle monolingue, même spécifique au domaine. Le lecteur peut donc se demander ce que donnerait un modèle multilingue pré-entraîné sur des données textuelles cliniques.

5.7 Conclusion

Cette thèse établit un lien entre les embeddings multilingues statiques et contextualisés, en constatant que l'alignement multilingue ciblé par les embeddings statiques est fortement corrélé avec les capacités de transfert translingue des embeddings contextualisés. Alors que les embeddings statiques peuvent également être formés de manière langage-indépendante, comme mBERT ou XLM-R, ces derniers modèles contextualisés offrent un meilleur alignement multilingue que les embeddings statiques qui ont été explicitement entraînés pour être alignés.

Les embeddings multilingues statiques, en particulier ceux non supervisés, souffrent de nombreuses limitations. Cette thèse montre que ces limitations ne se limitent pas à une méthode

[25] L'ensemble de données est disponible à l'adresse https://huggingface.co/datasets/Posos/MedNERF.
[26] car les modèles multilingues spécifiques au domaine sont rares

d'alignement spécifique, et toutes les méthodes testées qui reposent sur l'isométrie échouent à un certain degré concernant dans les cas limites : pour des langues éloignées et à faible ressource, et entre différents domaines.

En revanche, les modèles contextualisés semblent assez bien fonctionner tels quels. Bien que la littérature montre que certaines langues à faible ressource peuvent avoir des performances dégradées avec le transfert translingue, cette thèse suggère que le réalignement peut aider dans ce cas spécifique, à condition que les paires de mots utilisées pour le réalignement soient extraites avec précision, avec un dictionnaire bilingue plutôt qu'avec un outil d'alignement de mots.

De plus, lorsqu'ils sont utilisés en dehors de leur domaine de pré-entraînement, des Transformers multilingues comme mBERT peuvent toujours donner de bons résultats. Le dernier chapitre montre que de tels modèles offrent de bons résultats pour une tâche de NER clinique, comparables à ceux de méthodes basées sur la traduction.

Cette thèse valide que les embeddings contextualisés multilingues peuvent être utilisés dans un contexte clinique, en particulier dans un scénario multilingue sans adaptation préalable, où la sélection d'un modèle de traduction, et surtout d'alignement, peut devenir délicate. Cependant, le succès relatif de DrBERT Pubmed, un modèle pré-entraîné sur des textes biomédicaux et cliniques en anglais puis en français, plaide en faveur de futures recherches sur la pré-entraînement multilingue spécifique au domaine pour les modèles de langage.

Acronymes

BLI	Induction de Dictionaire Bilingue
CTL	Apprentissage par Transfert translingue
MLM	prédiction de mots masqués
mLM	modèle de langage multilingue
MA	alignement multilingue
WE	Word Embedding
SWE	word embedding statique
CWE	word embeding contextualisé
NLI	inférence de langage
NER	reconnaissance d'entités nommées
POS-tagging	étiquetage grammatical

Bibliography

Adebara, Ife, AbdelRahim Elmadany, Muhammad Abdul-Mageed, and Alcides Alcoba Inciarte (July 2023). "SERENGETI: Massively Multilingual Language Models for Africa." In: *Findings of the Association for Computational Linguistics: ACL 2023*. Toronto, Canada: Association for Computational Linguistics, pp. 1498–1537. URL: https://aclanthology.org/2023.findings-acl.97.

Ahuja, Kabir, Shanu Kumar, Sandipan Dandapat, and Monojit Choudhury (May 2022). "Multi Task Learning For Zero Shot Performance Prediction of Multilingual Models." In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, pp. 5454–5467. DOI: 10.18653/v1/2022.acl-long.374. URL: https://aclanthology.org/2022.acl-long.374.

Ahuja, Kabir et al. (2023). *MEGA: Multilingual Evaluation of Generative AI*. arXiv: 2303.12528 [cs.CL].

Alaux, Jean, Edouard Grave, Marco Cuturi, and Armand Joulin (2018). "Unsupervised Hyperalignment for Multilingual Word Embeddings." In: *CoRR* abs/1811.01124. arXiv: 1811.01124. URL: http://arxiv.org/abs/1811.01124.

Alvarez-Melis, David and Tommi Jaakkola (2018). "Gromov-Wasserstein Alignment of Word Embedding Spaces." In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, pp. 1881–1890. DOI: 10.18653/v1/D18-1214. URL: https://aclanthology.org/D18-1214.

Artetxe, Mikel, Gorka Labaka, and Eneko Agirre (July 2017). "Learning bilingual word embeddings with (almost) no bilingual data." In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, pp. 451–462. DOI: 10.18653/v1/P17-1042. URL: https://aclanthology.org/P17-1042.

— (July 2018a). "A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings." In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, pp. 789–798. DOI: 10.18653/v1/P18-1073. URL: https://aclanthology.org/P18-1073.

— (2018b). "Generalizing and improving bilingual word embedding mappings with a multi-step framework of linear transformations." In: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pp. 5012–5019.

Artetxe, Mikel, Sebastian Ruder, and Dani Yogatama (July 2020). "On the Cross-lingual Transferability of Monolingual Representations." In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 4623–4637. DOI: 10.18653/v1/2020.acl-main.421. URL: https://aclanthology.org/2020.acl-main.421.

Artetxe, Mikel and Holger Schwenk (2019). "Massively Multilingual Sentence Embeddings for Zero-Shot Cross-Lingual Transfer and Beyond." In: *Transactions of the Association for Computational Linguistics* 7, pp. 597–610. DOI: 10.1162/tacl_a_00288. URL: https://aclanthology.org/Q19-1038.

Artetxe, Mikel, Sebastian Ruder, Dani Yogatama, Gorka Labaka, and Eneko Agirre (July 2020). "A Call for More Rigor in Unsupervised Cross-lingual Learning." In: *Proceedings of the 58th*

Annual Meeting of the Association for Computational Linguistics. Online: Association for Computational Linguistics, pp. 7375–7388. DOI: 10.18653/v1/2020.acl-main.658. URL: https://aclanthology.org/2020.acl-main.658.

Artetxe, Mikel, Vedanuj Goswami, Shruti Bhosale, Angela Fan, and Luke Zettlemoyer (2023). *Revisiting Machine Translation for Cross-lingual Classification*. arXiv: 2305.14240 [cs.CL].

Bawden, Rachel et al. (Aug. 2019). "Findings of the WMT 2019 Biomedical Translation Shared Task: Evaluation for MEDLINE Abstracts and Biomedical Terminologies." In: *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*. Florence, Italy: Association for Computational Linguistics, pp. 29–53. DOI: 10.18653/v1/W19-5403. URL: https://aclanthology.org/W19-5403.

Bengio, Yoshua, Réjean Ducharme, and Pascal Vincent (2000). "A Neural Probabilistic Language Model." In: *Advances in Neural Information Processing Systems*. Ed. by T. Leen, T. Dietterich, and V. Tresp. Vol. 13. MIT Press. URL: https://proceedings.neurips.cc/paper/2000/file/728f206c2a01bf572b5940d7d9a8fa4c-Paper.pdf.

Bishara, Anthony J. and James B. Hittner (2017). "Confidence intervals for correlations when data are not normal." In: *Behavior Research Methods* 49.1, pp. 294–309. ISSN: 1554-3528. DOI: 10.3758/s13428-016-0702-8. URL: https://doi.org/10.3758/s13428-016-0702-8.

Bojanowski, Piotr, Edouard Grave, Armand Joulin, and Tomas Mikolov (2017). "Enriching Word Vectors with Subword Information." In: *Transactions of the Association for Computational Linguistics* 5, pp. 135–146. DOI: 10.1162/tacl_a_00051. URL: https://aclanthology.org/Q17-1010.

Borchert, Florian, Christina Lohr, Luise Modersohn, Jonas Witt, Thomas Langer, Markus Follmann, Matthias Gietzelt, Bert Arnrich, Udo Hahn, and Matthieu-P. Schapranow (June 2022). "GGPONC 2.0 - The German Clinical Guideline Corpus for Oncology: Curation Workflow, Annotation Policy, Baseline NER Taggers." In: *Proceedings of the Thirteenth Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, pp. 3650–3660. URL: https://aclanthology.org/2022.lrec-1.389.

Bressem, Keno K, Jens-Michalis Papaioannou, Paul Grundmann, Florian Borchert, Lisa C Adams, Leonhard Liu, Felix Busch, Lina Xu, Jan P Loyen, Stefan M Niehues, et al. (2023). "MEDBERT. de: A Comprehensive German BERT Model for the Medical Domain." In: *arXiv preprint arXiv:2303.08179*.

Brown, Tom et al. (2020). "Language Models are Few-Shot Learners." In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., pp. 1877–1901. URL: https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Cao, Steven, Nikita Kitaev, and Dan Klein (2020). "Multilingual Alignment of Contextual Word Representations." In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=r1xCMyBtPS.

Chazal, Frédéric, David Cohen-Steiner, Leonidas J. Guibas, Facundo Mémoli, and Steve Y. Oudot (2009). "Gromov-Hausdorff Stable Signatures for Shapes Using Persistence." In: *Proceedings of the Symposium on Geometry Processing*. Eurographics Association, 1393–1403.

Chen, Ting, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton (2020). "A Simple Framework for Contrastive Learning of Visual Representations." In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by Hal Daumé III and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, pp. 1597–1607. URL: https://proceedings.mlr.press/v119/chen20j.html.

Chung, Hyung Won, Thibault Fevry, Henry Tsai, Melvin Johnson, and Sebastian Ruder (2021). "Rethinking Embedding Coupling in Pre-trained Language Models." In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=xpFFI_NtgpW.

Clark, Kevin, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning (2020). "ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators." In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=r1xMH1BtvB.

Collobert, Ronan and Jason Weston (2008). "A unified architecture for natural language processing: deep neural networks with multitask learning." In: *International Conference on Machine Learning*.

Collobert, Ronan, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa (2011). "Natural Language Processing (Almost) from Scratch." In: *Journal of Machine Learning Research* 12.76, pp. 2493–2537. URL: http://jmlr.org/papers/v12/collobert11a.html.

Conneau, Alexis and Guillaume Lample (2019). "Cross-lingual Language Model Pretraining." In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett. Vol. 32. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper_files/paper/2019/file/c04c19c2c2474dbf5f7ac4372c5b9af1-Paper.pdf.

Conneau, Alexis, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel R. Bowman, Holger Schwenk, and Veselin Stoyanov (2018a). "XNLI: Evaluating Cross-lingual Sentence Representations." In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics.

Conneau, Alexis, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov (2018b). "XNLI: Evaluating Cross-lingual Sentence Representations." In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, pp. 2475–2485. DOI: 10.18653/v1/D18-1269. URL: https://aclanthology.org/D18-1269.

Conneau, Alexis, Shijie Wu, Haoran Li, Luke Zettlemoyer, and Veselin Stoyanov (July 2020a). "Emerging Cross-lingual Structure in Pretrained Language Models." In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 6022–6034. DOI: 10.18653/v1/2020.acl-main.536. URL: https://aclanthology.org/2020.acl-main.536.

Conneau, Alexis, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov (July 2020b). "Unsupervised Cross-lingual Representation Learning at Scale." In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 8440–8451. DOI: 10.18653/v1/2020.acl-main.747. URL: https://aclanthology.org/2020.acl-main.747.

Daras, Petros, Apostolos Axenopoulos, and Georgios Litos (2012). "Investigating the Effects of Multiple Factors Towards More Accurate 3-D Object Retrieval." In: *IEEE Transactions on Multimedia* 14.2, pp. 374–388. DOI: 10.1109/TMM.2011.2176111.

Deerwester, Scott C., Susan T. Dumais, Thomas K. Landauer, George W. Fumas, and L. L. Beck (1988). "Improving information retrieval using latent semantic indexing." In:

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (June 2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: https://aclanthology.org/N19-1423.

Dinu, Georgiana, Angeliki Lazaridou, and Marco Baroni (2015). *Improving zero-shot learning by mitigating the hubness problem*. arXiv: 1412.6568 [cs.CL].

Doddapaneni, Sumanth, Gowtham Ramesh, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh M. Khapra (2021). "A Primer on Pretrained Multilingual Language Models." In: *CoRR* abs/2107.00676. arXiv: 2107.00676.

Dossou, Bonaventure F. P., Atnafu Lambebo Tonja, Oreen Yousuf, Salomey Osei, Abigail Oppong, Iyanuoluwa Shode, Oluwabusayo Olufunke Awoyomi, and Chris Emezue (Dec. 2022). "AfroLM: A Self-Active Learning-based Multilingual Pretrained Language Model for 23 African Languages." In: *Proceedings of The Third Workshop on Simple and Efficient Natural Language Processing (SustaiNLP)*. Abu Dhabi, United Arab Emirates (Hybrid): Association for Computational Linguistics, pp. 52–64. URL: https://aclanthology.org/2022.sustainlp-1.11.

Dou, Zi-Yi and Graham Neubig (Apr. 2021). "Word Alignment by Fine-tuning Embeddings on Parallel Corpora." In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Online: Association for Computational Linguistics, pp. 2112–2128. DOI: 10.18653/v1/2021.eacl-main.181. URL: https://aclanthology.org/2021.eacl-main.181.

Du, Yunshu, Wojciech M. Czarnecki, Siddhant M. Jayakumar, Mehrdad Farajtabar, Razvan Pascanu, and Balaji Lakshminarayanan (2018). *Adapting Auxiliary Losses Using Gradient Similarity*. DOI: 10.48550/ARXIV.1812.02224. URL: https://arxiv.org/abs/1812.02224.

Dufter, Philipp and Hinrich Schütze (Nov. 2020). "Identifying Elements Essential for BERT's Multilinguality." In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, pp. 4423–4437. DOI: 10.18653/v1/2020.emnlp-main.358. URL: https://aclanthology.org/2020.emnlp-main.358.

Dušek, Ondřej, Jan Hajič, Jaroslava Hlaváčová, Jindřich Libovický, Pavel Pecina, Aleš Tamchyna, and Zdeňka Urešová (2017). *Khresmoi Summary Translation Test Data 2.0*. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University. URL: http://hdl.handle.net/11234/1-2122.

Dyer, Chris, Victor Chahuneau, and Noah A. Smith (June 2013). "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*. Atlanta, Georgia: Association for Computational Linguistics, pp. 644–648. URL: https://aclanthology.org/N13-1073.

Efimov, Pavel, Leonid Boytsov, Elena Arslanova, and Pavel Braslavski (2023). "The Impact of Cross-Lingual Adjustment of Contextual Word Representations on Zero-Shot Transfer." In: *Advances in Information Retrieval*. Cham: Springer Nature Switzerland, pp. 51–67. ISBN: 978-3-031-28241-6.

Efron, Bradley and Robert Tibshirani (1994). "An Introduction to the Bootstrap." In:

El Boukkouri, Hicham (Nov. 2021). "Domain adaptation of word embeddings through the exploitation of in-domain corpora and knowledge bases." Theses. Université Paris-Saclay. URL: https://theses.hal.science/tel-03560502.

Elazar, Yanai and Yoav Goldberg (2018). "Adversarial Removal of Demographic Attributes from Text Data." In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, pp. 11–21. DOI: 10.18653/v1/D18-1002. URL: https://aclanthology.org/D18-1002.

Firth, John R. (1959). *Studies in Linguistic Analysis*. DOI: 10.1086/464540.

Frei, Johann, Ludwig Frei-Stuber, and Frank Kramer (2022). "GERNERMED++: Transfer Learning in German Medical NLP." In: DOI: 10.48550/ARXIV.2206.14504. URL: https://arxiv.org/abs/2206.14504.

Glavaš, Goran, Robert Litschko, Sebastian Ruder, and Ivan Vulić (July 2019). "How to (Properly) Evaluate Cross-Lingual Word Embeddings: On Strong Baselines, Comparative Analyses, and Some Misconceptions." In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 710–721. DOI: 10.18653/v1/P19-1070. URL: https://aclanthology.org/P19-1070.

Goodfellow, Ian J., Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio (2014). "Generative Adversarial Nets." In: *NIPS*.

Gordon, Carolyn, David L. Webb, and Scott Wolpert (1992). *One cannot hear the shape of a drum*. arXiv: math/9207215 [math.DG].

Grave, Edouard, Armand Joulin, and Quentin Berthet (2018). "Unsupervised Alignment of Embeddings with Wasserstein Procrustes." In: *International Conference on Artificial Intelligence and Statistics*.

Gu, Yu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon (2021). "Domain-Specific Language Model Pretraining for Biomedical Natural Language Processing." In: 3.1. ISSN: 2691-1957. DOI: 10.1145/3458754. URL: https://doi.org/10.1145/3458754.

Hartmann, Mareike, Yova Kementchedjhieva, and Anders Søgaard (2018). "Why is unsupervised alignment of English embeddings from different algorithms so hard?" In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, pp. 582–586. DOI: 10.18653/v1/D18-1056. URL: https://aclanthology.org/D18-1056.

— (2019). "Comparing Unsupervised Word Translation Methods Step by Step." In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett. Vol. 32. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper_files/paper/2019/file/d15426b9c324676610fbb01360473ed8-Paper.pdf.

He, Pengcheng, Jianfeng Gao, and Weizhu Chen (2023). "DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing." In: *The Eleventh International Conference on Learning Representations*. URL: https://openreview.net/forum?id=sE7-XhLxHA.

He, Pengcheng, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen (2021). "{DEBERTA}: {DECODING}-{ENHANCED} {BERT} {WITH} {DISENTANGLED} {ATTENTION}." In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=XPZIaotutsD.

Henry, Sam, Kevin Buchan, Michele Filannino, Amber Stubbs, and Ozlem Uzuner (Oct. 2019). "2018 n2c2 shared task on adverse drug events and medication extraction in electronic health records." In: *Journal of the American Medical Informatics Association* 27.1, pp. 3–12. ISSN: 1527-974X. DOI: 10.1093/jamia/ocz166. eprint: https://academic.oup.com/jamia/article-pdf/27/1/3/34152182/ocz166.pdf. URL: https://doi.org/10.1093/jamia/ocz166.

Hoshen, Yedid and Lior Wolf (2018). "Non-Adversarial Unsupervised Word Translation." In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, pp. 469–478. DOI: 10.18653/v1/D18-1043. URL: https://aclanthology.org/D18-1043.

Houlsby, Neil, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly (2019). "Parameter-Efficient Transfer Learning for NLP." In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, pp. 2790–2799. URL: https://proceedings.mlr.press/v97/houlsby19a.html.

Hu, Junjie, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson (2020a). "XTREME: A Massively Multilingual Multi-Task Benchmark for Evaluating Cross-Lingual Generalization." In: *Proceedings of the 37th International Conference on Machine Learning*. ICML'20. JMLR.org.

— (2020b). "XTREME: A Massively Multilingual Multi-task Benchmark for Evaluating Cross-lingual Generalization." In: *CoRR* abs/2003.11080. arXiv: 2003.11080.

ImaniGooghari, Ayyoob et al. (July 2023). "Glot500: Scaling Multilingual Corpora and Language Models to 500 Languages." In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Toronto, Canada: Association for Computational Linguistics, pp. 1082–1117. URL: https://aclanthology.org/2023.acl-long.61.

Jiao, Xiaoqi, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu (Nov. 2020). "TinyBERT: Distilling BERT for Natural Language Understanding." In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, pp. 4163–4174. DOI: 10.18653/v1/2020.findings-emnlp.372. URL: https://aclanthology.org/2020.findings-emnlp.372.

Joshi, Mandar, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy (2020). "SpanBERT: Improving Pre-training by Representing and Predicting Spans." In: *Transactions of the Association for Computational Linguistics* 8, pp. 64–77. DOI: 10.1162/tacl_a_00300. URL: https://aclanthology.org/2020.tacl-1.5.

Joulin, Armand, Piotr Bojanowski, Tomas Mikolov, Hervé Jégou, and Edouard Grave (2018). "Loss in Translation: Learning Bilingual Word Mapping with a Retrieval Criterion." In: *Proceedings of*

the 2018 Conference on Empirical Methods in Natural Language Processing. Brussels, Belgium: Association for Computational Linguistics, pp. 2979–2984. DOI: 10.18653/v1/D18-1330. URL: https://aclanthology.org/D18-1330.

K, Karthikeyan, Zihan Wang, Stephen Mayhew, and Dan Roth (2020). "Cross-Lingual Ability of Multilingual BERT: An Empirical Study." In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=HJeT3yrtDr.

Kakwani, Divyanshu, Anoop Kunchukuttan, Satish Golla, Gokul N.C., Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar (Nov. 2020). "IndicNLPSuite: Monolingual Corpora, Evaluation Benchmarks and Pre-trained Multilingual Language Models for Indian Languages." In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, pp. 4948–4961. DOI: 10.18653/v1/2020.findings-emnlp.445. URL: https://aclanthology.org/2020.findings-emnlp.445.

Khattab, Omar and Matei Zaharia (2020). "ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT." In: *CoRR* abs/2004.12832. arXiv: 2004.12832. URL: https://arxiv.org/abs/2004.12832.

Koehn, Philipp (2005). "Europarl: A Parallel Corpus for Statistical Machine Translation." In: *Conference Proceedings: the tenth Machine Translation Summit*. AAMT. Phuket, Thailand: AAMT. URL: http://mt-archive.info/MTS-2005-Koehn.pdf.

Kondratyuk, Dan and Milan Straka (Nov. 2019). "75 Languages, 1 Model: Parsing Universal Dependencies Universally." In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pp. 2779–2795. DOI: 10.18653/v1/D19-1279. URL: https://aclanthology.org/D19-1279.

Kudo, Taku and John Richardson (Nov. 2018). "SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing." In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Brussels, Belgium: Association for Computational Linguistics, pp. 66–71. DOI: 10.18653/v1/D18-2012. URL: https://aclanthology.org/D18-2012.

Kulshreshtha, Saurabh, Jose Luis Redondo Garcia, and Ching-Yun Chang (Nov. 2020). "Cross-lingual Alignment Methods for Multilingual BERT: A Comparative Study." In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, pp. 933–942. DOI: 10.18653/v1/2020.findings-emnlp.83. URL: https://aclanthology.org/2020.findings-emnlp.83.

Labrak, Yanis, Adrien Bazoge, Richard Dufour, Mickael Rouvier, Emmanuel Morin, Béatrice Daille, and Pierre-Antoine Gourraud (July 2023). "DrBERT: A Robust Pre-trained Model in French for Biomedical and Clinical domains." In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Toronto, Canada: Association for Computational Linguistics, pp. 16207–16221. URL: https://aclanthology.org/2023.acl-long.896.

Lample, Guillaume, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou (2018). "Word translation without parallel data." In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=H196sainb.

Lan, Zhenzhong, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut (2020). "ALBERT: A Lite BERT for Self-supervised Learning of Language Representations." In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=H1eA7AEtvS.

Lauscher, Anne, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš (Nov. 2020). "From Zero to Hero: On the Limitations of Zero-Shot Language Transfer with Multilingual Transformers." In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, pp. 4483–4499. DOI: 10.18653/v1/2020.emnlp-main.363. URL: https://aclanthology.org/2020.emnlp-main.363.

Le, Hai Son, Alexandre Allauzen, and François Yvon (June 2012). "Continuous Space Translation Models with Neural Networks." In: *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Montréal, Canada: Association for Computational Linguistics, pp. 39–48. URL: https://aclanthology.org/N12-1005.

Lewis, Patrick, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk (July 2020). "MLQA: Evaluating Cross-lingual Extractive Question Answering." In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 7315–7330. DOI: 10.18653/v1/2020.acl-main.653. URL: https://aclanthology.org/2020.acl-main.653.

Li, Tianjian and Kenton Murray (July 2023). "Why Does Zero-Shot Cross-Lingual Generation Fail? An Explanation and a Solution." In: *Findings of the Association for Computational Linguistics: ACL 2023*. Toronto, Canada: Association for Computational Linguistics, pp. 12461–12476. URL: https://aclanthology.org/2023.findings-acl.789.

Libovický, Jindřich, Rudolf Rosa, and Alexander Fraser (Nov. 2020). "On the Language Neutrality of Pre-trained Multilingual Representations." In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, pp. 1663–1674. DOI: 10.18653/v1/2020.findings-emnlp.150. URL: https://aclanthology.org/2020.findings-emnlp.150.

Liebel, Lukas and Marco Körner (2018). "Auxiliary Tasks in Multi-task Learning." In: *CoRR* abs/1805.06334. arXiv: 1805.06334. URL: http://arxiv.org/abs/1805.06334.

Liu, Chi-Liang, Tsung-Yuan Hsu, Yung-Sung Chuang, and Hung-Yi Lee (2020a). *A Study of Cross-Lingual Ability and Language-specific Information in Multilingual BERT*. arXiv: 2004.09205 [cs.CL].

Liu, Fenglin, Meng Gao, Yuanxin Liu, and Kai Lei (Nov. 2019a). "Self-Adaptive Scaling for Learnable Residual Structure." In: *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*. Hong Kong, China: Association for Computational Linguistics, pp. 862–870. DOI: 10.18653/v1/K19-1080. URL: https://aclanthology.org/K19-1080.

Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov (2019b). "RoBERTa: A Robustly Optimized BERT Pretraining Approach." In: *CoRR* abs/1907.11692. arXiv: 1907.11692. URL: http://arxiv.org/abs/1907.11692.

Liu, Yinhan, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer (2020b). "Multilingual Denoising Pre-training for Neural Machine Translation." In: *Transactions of the Association for Computational Linguistics* 8, pp. 726–742. DOI: 10.1162/tacl_a_00343. URL: https://aclanthology.org/2020.tacl-1.47.

Loshchilov, Ilya and Frank Hutter (2019). "Decoupled Weight Decay Regularization." In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=Bkg6RiCqY7.

Maaten, Laurens van der and Geoffrey Hinton (2008). "Visualizing Data using t-SNE." In: *Journal of Machine Learning Research* 9.86, pp. 2579–2605. URL: http://jmlr.org/papers/v9/vandermaaten08a.html.

Marchisio, Kelly, Neha Verma, Kevin Duh, and Philipp Koehn (Dec. 2022). "IsoVec: Controlling the Relative Isomorphism of Word Embedding Spaces." In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, pp. 6019–6033. URL: https://aclanthology.org/2022.emnlp-main.404.

Martin, Louis, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric de la Clergerie, Djamé Seddah, and Benoît Sagot (July 2020). "CamemBERT: a Tasty French Language Model." In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 7203–7219. DOI: 10.18653/v1/2020.acl-main.645. URL: https://aclanthology.org/2020.acl-main.645.

Miceli Barone, Antonio Valerio (Aug. 2016). "Towards cross-lingual distributed representations without parallel text trained with adversarial autoencoders." In: *Proceedings of the 1st Workshop on Representation Learning for NLP*. Berlin, Germany: Association for Computational Linguistics, pp. 121–126. DOI: 10.18653/v1/W16-1614. URL: https://aclanthology.org/W16-1614.

Mikolov, Tomas, Quoc V. Le, and Ilya Sutskever (2013). "Exploiting Similarities among Languages for Machine Translation." In: *ArXiv* abs/1309.4168.

Mikolov, Tomas, Wen-tau Yih, and Geoffrey Zweig (June 2013). "Linguistic Regularities in Continuous Space Word Representations." In: *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Atlanta, Georgia: Association for Computational Linguistics, pp. 746–751. URL: https://aclanthology.org/N13-1090.

Mikolov, Tomas, Kai Chen, Gregory S. Corrado, and Jeffrey Dean (2013). "Efficient Estimation of Word Representations in Vector Space." In: *International Conference on Learning Representations*.

Muennighoff, Niklas et al. (July 2023). "Crosslingual Generalization through Multitask Finetuning." In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Toronto, Canada: Association for Computational Linguistics, pp. 15991–16111. URL: https://aclanthology.org/2023.acl-long.891.

Névéol, Aurélie, Cyril Grouin, Jérémy Leixa, Sophie Rosset, and Pierre Zweigenbaum (2014). "The Quaero French Medical Corpus : A Ressource for Medical Entity Recognition and Normalization." In:

Ng, Nathan, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov (Aug. 2019). "Facebook FAIR's WMT19 News Translation Task Submission." In: *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*. Florence, Italy: Association for Computational Linguistics, pp. 314–319. DOI: 10.18653/v1/W19-5333. URL: https://aclanthology.org/W19-5333.

— (2020). "Facebook FAIR's WMT19 News Translation Task Submission." In: *Proc. of WMT*.

Ogueji, Kelechi, Yuxin Zhu, and Jimmy Lin (Nov. 2021). "Small Data? No Problem! Exploring the Viability of Pretrained Multilingual Language Models for Low-resourced Languages." In: *Proceedings of the 1st Workshop on Multilingual Representation Learning*. Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 116–126. DOI: 10.18653/v1/2021.mrl-1.11. URL: https://aclanthology.org/2021.mrl-1.11.

Ouyang, Long et al. (2022). *Training language models to follow instructions with human feedback*. DOI: 10.48550/ARXIV.2203.02155. URL: https://arxiv.org/abs/2203.02155.

Pan, Lin, Chung-Wei Hang, Haode Qi, Abhishek Shah, Saloni Potdar, and Mo Yu (June 2021). "Multilingual BERT Post-Pretraining Alignment." In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, pp. 210–219. DOI: 10.18653/v1/2021.naacl-main.20. URL: https://aclanthology.org/2021.naacl-main.20.

Pan, Xiaoman, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji (July 2017). "Cross-lingual Name Tagging and Linking for 282 Languages." In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, pp. 1946–1958. DOI: 10.18653/v1/P17-1178. URL: https://aclanthology.org/P17-1178.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu (July 2002). "Bleu: a Method for Automatic Evaluation of Machine Translation." In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, pp. 311–318. DOI: 10.3115/1073083.1073135. URL: https://aclanthology.org/P02-1040.

Patra, Barun, Joel Ruben Antony Moniz, Sarthak Garg, Matthew R. Gormley, and Graham Neubig (July 2019). "Bilingual Lexicon Induction with Semi-supervision in Non-Isometric Embedding Spaces." In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 184–193. DOI: 10.18653/v1/P19-1018. URL: https://aclanthology.org/P19-1018.

Peters, Matthew E., Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer (June 2018). "Deep Contextualized Word Representations." In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational*

Linguistics: Human Language Technologies, Volume 1 (Long Papers). New Orleans, Louisiana: Association for Computational Linguistics, pp. 2227–2237. DOI: 10.18653/v1/N18-1202. URL: https://aclanthology.org/N18-1202.

Pfeiffer, Jonas, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder (Nov. 2020). "MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer." In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, pp. 7654–7673. DOI: 10.18653/v1/2020.emnlp-main.617. URL: https://aclanthology.org/2020.emnlp-main.617.

Pfeiffer, Jonas, Francesco Piccinno, Massimo Nicosia, Xinyi Wang, Machel Reid, and Sebastian Ruder (2023). *mmT5: Modular Multilingual Pre-Training Solves Source Language Hallucinations*. arXiv: 2305.14224 [cs.CL].

Philippy, Fred, Siwen Guo, and Shohreh Haddadan (July 2023). "Towards a Common Understanding of Contributing Factors for Cross-Lingual Transfer in Multilingual Language Models: A Review." In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Toronto, Canada: Association for Computational Linguistics, pp. 5877–5891. URL: https://aclanthology.org/2023.acl-long.323.

Pires, Telmo, Eva Schlinger, and Dan Garrette (July 2019). "How Multilingual is Multilingual BERT?" In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 4996–5001. DOI: 10.18653/v1/P19-1493. URL: https://aclanthology.org/P19-1493.

Radford, Alec and Karthik Narasimhan (2018). "Improving Language Understanding by Generative Pre-Training." In:

Raffel, Colin, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu (2020). "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer." In: *Journal of Machine Learning Research* 21.140, pp. 1–67. URL: http://jmlr.org/papers/v21/20-074.html.

Ramponi, Alan and Barbara Plank (Dec. 2020). "Neural Unsupervised Domain Adaptation in NLP—A Survey." In: *Proceedings of the 28th International Conference on Computational Linguistics*. Barcelona, Spain (Online): International Committee on Computational Linguistics, pp. 6838–6855. DOI: 10.18653/v1/2020.coling-main.603. URL: https://aclanthology.org/2020.coling-main.603.

Rebuffi, Sylvestre-Alvise, Hakan Bilen, and Andrea Vedaldi (2017). "Learning multiple visual domains with residual adapters." In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Vol. 30. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/e7b24b112a44fdd9ee93bdf998c6ca0e-Paper.pdf.

Rei, Ricardo, Craig Stewart, Ana C Farinha, and Alon Lavie (Nov. 2020). "COMET: A Neural Framework for MT Evaluation." In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, pp. 2685–2702. DOI: 10.18653/v1/2020.emnlp-main.213. URL: https://aclanthology.org/2020.emnlp-main.213.

Reimers, Nils and Iryna Gurevych (Nov. 2019). "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks." In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pp. 3982–3992. DOI: 10.18653/v1/D19-1410. URL: https://aclanthology.org/D19-1410.

Rönnqvist, Samuel, Valtteri Skantsi, Miika Oinonen, and Veronika Laippala (2021). "Multilingual and Zero-Shot is Closing in on Monolingual Web Register Classification." In: *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*. Reykjavik, Iceland (Online): Linköping University Electronic Press, Sweden, pp. 157–165. URL: https://aclanthology.org/2021.nodalida-main.16.

Roy, Uma, Noah Constant, Rami Al-Rfou, Aditya Barua, Aaron Phillips, and Yinfei Yang (Nov. 2020). "LAReQA: Language-Agnostic Answer Retrieval from a Multilingual Pool." In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online:

Association for Computational Linguistics, pp. 5919–5930. DOI: 10.18653/v1/2020.emnlp-main.477. URL: https://aclanthology.org/2020.emnlp-main.477.

Ruscio, John (2008). "Constructing Confidence Intervals for Spearman's Rank Correlation with Ordinal Data: A Simulation Study Comparing Analytic and Bootstrap Methods." In: *Journal of Modern Applied Statistical Methods* 7, p. 7.

Rust, Phillip, Jonas Pfeiffer, Ivan Vulić, Sebastian Ruder, and Iryna Gurevych (Aug. 2021). "How Good is Your Tokenizer? On the Monolingual Performance of Multilingual Language Models." In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, pp. 3118–3135. DOI: 10.18653/v1/2021.acl-long.243. URL: https://aclanthology.org/2021.acl-long.243.

Sanh, Victor, Lysandre Debut, Julien Chaumond, and Thomas Wolf (2019). "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter." In: *CoRR* abs/1910.01108. arXiv: 1910.01108. URL: http://arxiv.org/abs/1910.01108.

Scheible, Raphael, Fabian Thomczyk, Patric Tippmann, Victor Jaravine, and Martin Boeker (2020). "GottBERT: a pure German language model." In: *arXiv preprint arXiv:2012.02110*.

Schönemann, Peter H. (1966). "A generalized solution of the orthogonal procrustes problem." In: *Psychometrika* 31.1, pp. 1–10. ISSN: 1860-0980. URL: https://doi.org/10.1007/BF02289451.

Schwenk, Holger (2007). "Continuous space language models." In: *Comput. Speech Lang.* 21, pp. 492–518.

Singh, Jasdeep, Bryan McCann, Richard Socher, and Caiming Xiong (Nov. 2019). "BERT is Not an Interlingua and the Bias of Tokenization." In: *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*. Hong Kong, China: Association for Computational Linguistics, pp. 47–55. DOI: 10.18653/v1/D19-6106. URL: https://aclanthology.org/D19-6106.

Søgaard, Anders, Sebastian Ruder, and Ivan Vulić (July 2018). "On the Limitations of Unsupervised Bilingual Dictionary Induction." In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, pp. 778–788. DOI: 10.18653/v1/P18-1072. URL: https://aclanthology.org/P18-1072.

Sogaard, Anders, Manaal Faruqui, Ivan Vulic, and Sebastian Ruder (2019). *Cross-Lingual Word Embeddings*. Morgan & Claypool Publishers. ISBN: 1681730634.

Solaiman, Irene, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, and Jasmine Wang (2019). "Release Strategies and the Social Impacts of Language Models." In: *CoRR* abs/1908.09203. arXiv: 1908.09203. URL: http://arxiv.org/abs/1908.09203.

Srinivasan, Anirudh, Sunayana Sitaram, Tanuja Ganu, Sandipan Dandapat, Kalika Bali, and Monojit Choudhury (2021). *Predicting the Performance of Multilingual NLP Models*. arXiv: 2110.08875 [cs.CL].

Tanti, Marc, Lonneke van der Plas, Claudia Borg, and Albert Gatt (Nov. 2021). "On the Language-specificity of Multilingual BERT and the Impact of Fine-tuning." In: *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*. Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 214–227. DOI: 10.18653/v1/2021.blackboxnlp-1.15. URL: https://aclanthology.org/2021.blackboxnlp-1.15.

Tenney, Ian, Dipanjan Das, and Ellie Pavlick (July 2019). "BERT Rediscovers the Classical NLP Pipeline." In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 4593–4601. DOI: 10.18653/v1/P19-1452. URL: https://aclanthology.org/P19-1452.

The GUDHI Project (2020). *GUDHI User and Reference Manual*. 3.4.0. GUDHI Editorial Board. URL: https://gudhi.inria.fr/doc/3.4.0/.

Tiedemann, Jörg and Santhosh Thottingal (Nov. 2020a). "OPUS-MT – Building open translation services for the World." In: *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*. Lisboa, Portugal: European Association for Machine Translation, pp. 479–480. URL: https://aclanthology.org/2020.eamt-1.61.

Tiedemann, Jörg and Santhosh Thottingal (2020b). "OPUS-MT — Building open translation services for the World." In: *Proceedings of the 22nd Annual Conferenec of the European Association for Machine Translation (EAMT)*. Lisbon, Portugal.

Turian, Joseph, Lev-Arie Ratinov, and Yoshua Bengio (July 2010). "Word Representations: A Simple and General Method for Semi-Supervised Learning." In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Uppsala, Sweden: Association for Computational Linguistics, pp. 384–394. URL: https://aclanthology.org/P10-1040.

Turney, Peter D. and Patrick Pantel (2010). "From Frequency to Meaning: Vector Space Models of Semantics." In: *J. Artif. Int. Res.*

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin (2017). "Attention Is All You Need." In: *CoRR* abs/1706.03762. arXiv: 1706.03762. URL: http://arxiv.org/abs/1706.03762.

Vogel, Stephan, Hermann Ney, and Christoph Tillmann (1996). "HMM-Based Word Alignment in Statistical Translation." In: *COLING 1996 Volume 2: The 16th International Conference on Computational Linguistics*. URL: https://aclanthology.org/C96-2141.

Vu, Tu, Aditya Barua, Brian Lester, Daniel Cer, Mohit Iyyer, and Noah Constant (Dec. 2022). "Overcoming Catastrophic Forgetting in Zero-Shot Cross-Lingual Generation." In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, pp. 9279–9300. URL: https://aclanthology.org/2022.emnlp-main.630.

Vulić, Ivan, Goran Glavaš, Roi Reichart, and Anna Korhonen (Nov. 2019). "Do We Really Need Fully Unsupervised Cross-Lingual Embeddings?" In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pp. 4407–4418. DOI: 10.18653/v1/D19-1449. URL: https://aclanthology.org/D19-1449.

Wang, Kexin, Nils Reimers, and Iryna Gurevych (Nov. 2021). "TSDAE: Using Transformer-based Sequential Denoising Auto-Encoderfor Unsupervised Sentence Embedding Learning." In: *Findings of the Association for Computational Linguistics: EMNLP 2021*. Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 671–688. DOI: 10.18653/v1/2021.findings-emnlp.59. URL: https://aclanthology.org/2021.findings-emnlp.59.

Wang, Wenhui, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou (2020). "MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers." In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., pp. 5776–5788. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Wang, Yuxuan, Wanxiang Che, Jiang Guo, Yijia Liu, and Ting Liu (Nov. 2019). "Cross-Lingual BERT Transformation for Zero-Shot Dependency Parsing." In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pp. 5721–5727. DOI: 10.18653/v1/D19-1575. URL: https://aclanthology.org/D19-1575.

Warstadt, Alex, Amanpreet Singh, and Samuel R. Bowman (2019). "Neural Network Acceptability Judgments." In: *Transactions of the Association for Computational Linguistics* 7, pp. 625–641. DOI: 10.1162/tacl_a_00290. URL: https://aclanthology.org/Q19-1040.

Wikimedia Foundation (2019). "ACL 2019 Fourth Conference on Machine Translation (WMT19), Shared Task: Machine Translation of News." In: URL: http://www.statmt.org/wmt19/translation-task.html.

Wolf, Thomas et al. (Oct. 2020). "Transformers: State-of-the-Art Natural Language Processing." In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, pp. 38–45. URL: https://www.aclweb.org/anthology/2020.emnlp-demos.6.

Wu, Shijie and Mark Dredze (Nov. 2019). "Beto, Bentz, Becas: The Surprising Cross-Lingual Effectiveness of BERT." In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pp. 833–844. DOI: 10.18653/v1/D19-1077. URL: https://aclanthology.org/D19-1077.

— (Nov. 2020). "Do Explicit Alignments Robustly Improve Multilingual Encoders?" In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, pp. 4471–4482. DOI: 10.18653/v1/2020.emnlp-main.362. URL: https://aclanthology.org/2020.emnlp-main.362.

Wu, Yonghui et al. (2016). *Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation*. arXiv: 1609.08144 [cs.CL].

Xue, Linting, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel (June 2021). "mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer." In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, pp. 483–498. DOI: 10.18653/v1/2021.naacl-main.41. URL: https://aclanthology.org/2021.naacl-main.41.

Yang, Jian, Shuming Ma, Dongdong Zhang, ShuangZhi Wu, Zhoujun Li, and Ming Zhou (2020). "Alternating Language Modeling for Cross-Lingual Pre-Training." In: *Proceedings of the AAAI Conference on Artificial Intelligence* 34.05, pp. 9386–9393. DOI: 10.1609/aaai.v34i05.6480. URL: https://ojs.aaai.org/index.php/AAAI/article/view/6480.

Yarmohammadi, Mahsa et al. (Nov. 2021). "Everything Is All It Takes: A Multipronged Strategy for Zero-Shot Cross-Lingual Information Extraction." In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 1950–1967. DOI: 10.18653/v1/2021.emnlp-main.149. URL: https://aclanthology.org/2021.emnlp-main.149.

Zeman, Daniel et al. (2020). *Universal Dependencies 2.6*. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University. URL: http://hdl.handle.net/11234/1-3226.

Zhang, Biao, Philip Williams, Ivan Titov, and Rico Sennrich (July 2020). "Improving Massively Multilingual Neural Machine Translation and Zero-Shot Translation." In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 1628–1639. DOI: 10.18653/v1/2020.acl-main.148. URL: https://aclanthology.org/2020.acl-main.148.

Zhang, Meng, Yang Liu, Huanbo Luan, and Maosong Sun (July 2017a). "Adversarial Training for Unsupervised Bilingual Lexicon Induction." In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, pp. 1959–1970. DOI: 10.18653/v1/P17-1179. URL: https://aclanthology.org/P17-1179.

— (Sept. 2017b). "Earth Mover's Distance Minimization for Unsupervised Bilingual Lexicon Induction." In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, pp. 1934–1945. DOI: 10.18653/v1/D17-1207. URL: https://aclanthology.org/D17-1207.

Zhang, Sheng, Kevin Duh, and Benjamin Van Durme (June 2018). "Fine-grained Entity Typing through Increased Discourse Context and Adaptive Classification Thresholds." In: *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*. New Orleans, Louisiana: Association for Computational Linguistics, pp. 173–179. DOI: 10.18653/v1/S18-2022. URL: https://aclanthology.org/S18-2022.

Zhao, Wei, Steffen Eger, Johannes Bjerva, and Isabelle Augenstein (Aug. 2021). "Inducing Language-Agnostic Multilingual Representations." In: *Proceedings of *SEM 2021: The Tenth Joint Conference on Lexical and Computational Semantics*. Online: Association for Computational Linguistics, pp. 229–240. DOI: 10.18653/v1/2021.starsem-1.22. URL: https://aclanthology.org/2021.starsem-1.22.

Zhou, Chunting et al. (2023). *LIMA: Less Is More for Alignment*. arXiv: 2305.11206 [cs.CL].

Zweigenbaum, Pierre, Serge Sharoff, and Reinhard Rapp (Aug. 2017). "Overview of the Second BUCC Shared Task: Spotting Parallel Sentences in Comparable Corpora." In: *Proceedings of the 10th Workshop on Building and Using Comparable Corpora*. Vancouver, Canada: Association for Computational Linguistics, pp. 60–67. DOI: 10.18653/v1/W17-2512. URL: https://aclanthology.org/W17-2512.

— (2018). "Overview of the Third BUCC Shared Task: Spotting Parallel Sentences in Comparable Corpora." In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.