Generative Probabilistic Alignment Models for Words and Subwords: a Systematic Exploration of the Limits and Potentials of Neural Parametrizations

Anh Khoa Ngo Ho

Generative Probabilistic Alignment Models for Words and Subwords: a Systematic Exploration of the Limits and Potentials of Neural Parametrizations

Thèse de doctorat de l'université Paris-Saclay

École doctorale n° 580 Sciences et technologies de l'information et de la communication (ED STIC)
Spécialité de doctorat: Informatique
Unité de recherche: Université Paris-Saclay, CNRS, Laboratoire interdisciplinaire des sciences du numérique, 91405, Orsay, France
Référent: Faculté des sciences d'Orsay

Thèse présentée et soutenue à Orsay, le 09 février 2021, par

Anh Khoa NGO HO

Composition du jury:

Pierre Zweigenbaum Directeur de Recherche, CNRS (LISN), Université Paris-Saclay	Président
Loïc Barrault Maître de Conférence, Université de Sheffield	Rapporteur & Examinateur
Yves Lepage Professeur, Université de Waseda	Rapporteur & Examinateur
Nadi Tomeh Maître de Conférence, Université Sorbonne Paris Nord	Examinateur
François Yvon Directeur de Recherche, CNRS (LISN), Université Paris-Saclay	Directeur de thèse

Thèse de doctorat

NNT: 2021UPASG014

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre: Modèles d'Alignement Probabilistes Génératifs pour les Mots et Sous-mots: une Exploration Systématique des Limites et Potentialités des Paramétrisations Neuronales

Mots clés: Traduction automatique, Alignement de mots, Réseaux de neurones artificiels

Résumé: L'alignement consiste à mettre en correspondance des unités au sein de bitextes, associant un texte en langue source et sa traduction dans une langue cible. L'alignement peut se concevoir à plusieurs niveaux: entre phrases, entre groupes de mots, entre mots, voire à un niveau plus fin lorsque l'une des langues est morphologiquement complexe, ce qui implique d'aligner des fragments de mot (morphèmes). L'alignement peut être envisagé également sur des structures linguistiques plus complexes des arbres ou des graphes. Il s'agit d'une tâche complexe, sous-spécifiée, que les humains réalisent avec difficulté. Son automatisation est un problème exemplaire du traitement des langues, historiquement associé aux premiers modèles de traduction probabilistes. L'arrivée à maturité de nouveaux modèles pour le traitement automatique des langues, reposant sur des représentations vectorielles calculées par des réseaux de neurones permet de reposer la question du calcul de ces alignements. Cette recherche vise donc à concevoir des modèles neuronaux susceptibles d'être appris sans supervision pour dépasser certaines des limitations des modèles d'alignement statistique et améliorer l'état de l'art en matière de précision des alignements automatiques.

Title: Generative Probabilistic Alignment Models for Words and Subwords: a Systematic Exploration of the Limits and Potentials of Neural Parametrizations

Keywords: Machine translation, Word alignment, Artificial neural network

Abstract: Alignment consists of establishing a mapping between units in a bitext, combining a text in a source language and its translation in a target language. Alignments can be computed at several levels: between documents, between sentences, between phrases, between words, or even between smaller units when one of the languages is morphologically complex, which implies to align fragments of words (morphemes). Alignments can also be considered between more complex linguistic structures such as trees or graphs. This is a complex, underspecified task that humans accomplish with difficulty. Its automation is a notoriously difficult problem in natural language processing, historically associated with the first probabilistic word-based translation models. The design of new models for natural language processing, based on distributed representations computed by neural networks, allows us to question and revisit the computation of these alignments. This research, therefore, aims to comprehensively understand the limitations of existing statistical alignment models and to design neural models that can be learned without supervision to overcome these drawbacks and to improve the state of art in terms of alignment accuracy.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France

Acknowledgments

I would like to express my deepest and most sincere gratitude to my thesis director François Yvon for his inspiration, patience and encouragement during the past four years. He guided me to explore an interesting domain of research with a host of challenges: Bitext alignment. His precious knowledge, great ideas and unfailing support helped me finish this thesis. I could not have imagined having a better adviser and mentor for my Ph.D study.

I also would like to warmly thank the members of the jury, Pierre Zweigenbaum, Loïc Barrault, Yves Lepage and Nadi Tomeh for their questions, comments and also corrections for my thesis on the day of my defense.

I would like to show my gratitude to the great people at LIMSI: Alexandre Allauzen, Laurence Rostaing, Jean-Claude Barbet, Pascal Desroches, Sophie Pageau-Maurice, Nicolas Rajaratnam, Jean-Luc Gauvain, Gilles Adda, William Smondack... I would like to thank to Stéphanie Druetta and Anne Vilnat at EDSTIC, Vanessa Delaisse and Laurie Vincent at HR department. They kindly helped me to do my thesis procedure paperwork.

I would like to thank to my lab-mates Aina, Aman, Benjamin, Charlotte, Franck, Jitao, Julia, Lauriane, Léo, Marc, François, Margot, Matthieu, Paul, Pierre, Pooyan, Quang, Rachel, Ruiqing, Shu, Soyoung, Syrielle, Xinneng, Yuming ...

Finally, I am deeply grateful to my parents, my brother, my sisters and my friends in Vietnam and in France for their endless support. I would like to thank to Ha Phuong Nguyen, Vo Linh Lan, Vu Trong Bach, Nguyen Lam Phuc Thinh, Nguyen Ly Bao Duy ... They are always there for me and honestly believe in what I am doing.

Contents

List of Figures

List of Tables

Acronyms

NLP Natural Language Processing

SMT Statistical Machine Translation

BPE Byte Pair Encoding

EM Expectation-Maximization

PoS Part-of-Speeches
PR Posterior Regularization

AER Alignment Error Rate

F1 F-score

ACC Accuracy

PRE Precision

REC Recall

TP True Positive

FP False Positive
FP True Negative
FP False Negative

En English

XX Foreign language

HMM Hidden Markov Model

NN Neural Network

RNN Recurrent Neural Network

LSTM Long Short-Term Memory

CNN Convolutional Neural Network

KL Kullback-Leibler divergence

UNK Unknown word

Chapter 1

Introduction

Research in natural language processing (NLP) are nowadays in quest of analyzing successfully large amounts of natural language data, by using the power of artificial intelligence systems. The applications of NLP range from spoken language, such as identifying and transforming it into text by computers (automatic speech recognition and language understanding), to language interpretation (machine translation, information extraction, automated reasoning, question-answering and text categorization). An important supporting task for machine translation is Bitext alignment [Tiedemann, 2011] consisting of establishing a mapping between units in a collection of parallel texts, combining a text in a source language and its translation in a target language. Alignments can be computed at multiple levels: between *documents* [Resnik, 1999, Fung and Cheung, 2004a, Paetzold et al., 2017], sentences [Brown et al., 1991, Melamed, 1996b, Schwenk, 2018], *phrases* [Och and Weber, 1998, Wisniewski et al., 2010, Nishino et al., 2016], *words* [Vogel et al., 1996, Melamed, 2000, Och and Ney, 2003, Sabet et al., 2020], or even between *smaller units* [Garg et al., 2019] when one of the languages is morphologically complex.

Bitext alignments at different levels of granularities have a very broad range of uses [Véronis, 2000]. Bitext corpora support human and machine translation. Statistical and example-based machine translation systems [Nagao, 1984] use them to obtain chunk alignments and to derive the parameters for their statistical models. Translation memories and computer-aided translation tools use alignment to extract domain-specific terminologies [Langlais et al., 2000, Kwong et al., 2002, Bourdaillet et al., 2009, Esplà-Gomis et al., 2011, Pham et al., 2018]. Neural machine translation (NMT) systems can enforce constraints in decoding by using alignment (e.g., coverage constraints [Tu et al., 2016a,b]). These systems also benefit from explicit alignments, which explain the translation predictions [Stahlberg et al., 2018]. Explainable and interpretable systems for NMT and also for machine learning attract more and more attention in the research community [Karpathy et al., 2015, Li et al., 2016, Ribeiro et al., 2016, Doshi-Velez and Kim, 2017, Alvarez-Melis and Jaakkola, 2017, Ding et al., 2017, Feng et al., 2018].

In addition, aligned bitexts assist in building bilingual dictionary [Melamed, 1996a], cross-language information retrieval [Wang and Oard, 2005], cross-lingual syntactic learning [Yarowsky et al., 2001, Smith and Smith, 2004, Hwa et al., 2005], query expansion in monolingual information retrieval [Xu et al., 2002, Riezler et al., 2007], synonym acquisition [van der Plas and Tiedemann, 2006], paraphrases [Pang et al., 2003, Quirk et al., 2004, Bannard and Callison-Burch, 2005], word sense disambiguation [Resnik, 1997]. Bitext corpora provide better interfaces for lexicographers, annotators and translators [Klavans and Tzoukermann, 1990], and also better tools for foreign language learners and bilingual readers [Yvon et al., 2016].

Word alignment is a fundamental step in extracting translation information from parallel sentences. It helps to determine which words in the source sentence correspond to which words in the target sentence. The information that can be extracted from such texts is bilingual dictionaries, transfer rules, and information about word order differences between languages. It is also useful for other applications such as translation memory cleaning and machine translation. This task can be done based on a large word-aligned corpus. However, it is not an easy task to

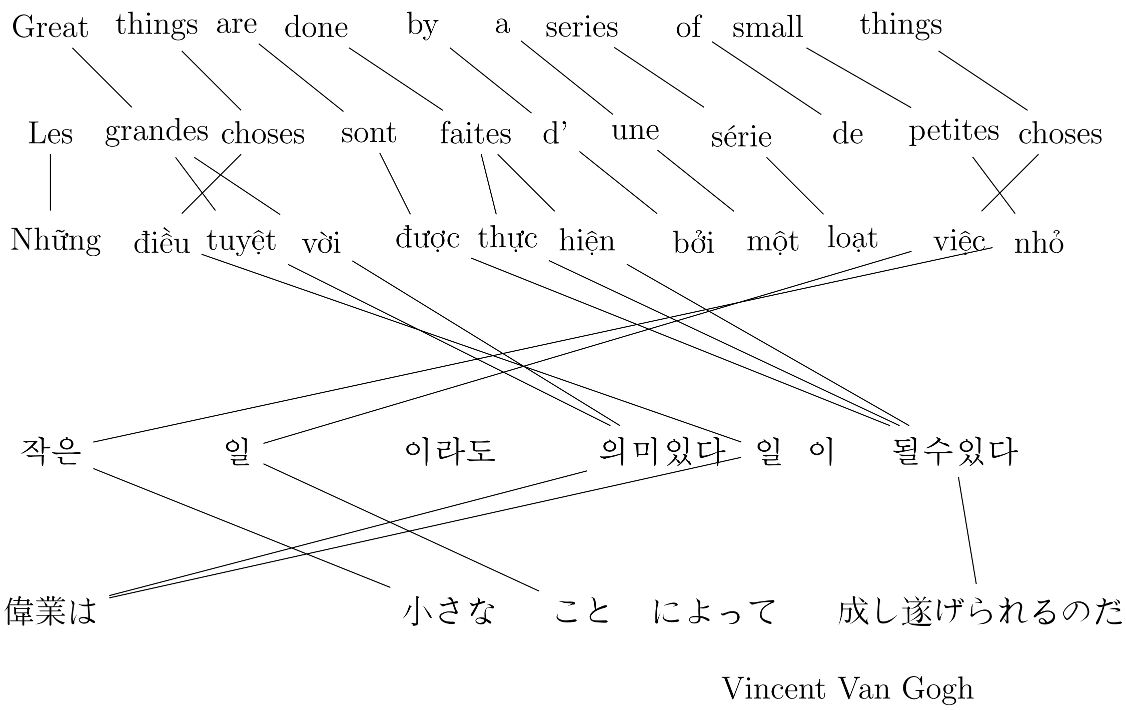


Figure 1.1: Difficulties in word alignment for English, French, Vietnamese, Korean and Japanese: Should "Les" align with "things" ? Should "faites" align with " được"? Should "de" align with " loạt" or " việc"? How to process the unaligned words ?

decide which source and target words correspond in a parallel text (see Figure 1.1) and manual word alignment can be very time-consuming. Until recently, the most predominant automatic word alignment models were statistical, as represented by the IBM Models of Brown et al. [1993b] and the HMM model of Vogel et al. [1996]. However, the quality of automatic word alignment computed by such models is far from perfect, especially if parallel data is scarce (see Figure 1.2). These models are typically challenged by low-frequency words, whose co-occurrences are poorly estimated and they also fail to take into account context information in alignment. Moreover, they are based on strict assumptions that make them unable to generate natural translations as they can only perform asymmetrical alignments. The design of new models for NLP, based on distributed representations computed by neural networks, allows us to question and revisit the computation of these alignments.

We also see that neural networks demonstrate state-of-the-art performance for a wide range of NLP areas such as text classification [Kim, 2014, Zhang and LeCun, 2015], named entity recognition [Lample et al., 2016, Wang et al., 2015a], semantic parsing [Yih et al., 2015], paraphrase detection [Bogdanova et al., 2015], language generation [Garbacea and Mei, 2020], speech recognition [Abdel-Hamid et al., 2014], character recognition [Memon et al., 2020], spell checking [Etoori et al., 2018] and especially machine translation [Cho et al., 2014a, Bahdanau et al., 2015, Luong et al., 2015]. Our main question is if neural networks bring state-of-the-art performance to the word alignment task.

This thesis aims to design neural models that can be learned without supervision to overcome some of the limitations of existing statistical alignment models and to improve the state of the art in terms of alignment accuracy. Moreover, we also need a collection of statistical tools to comprehensively observe these limitations and also the improvements of these neural models. Note that this dissertation is completed with a companion document [Ngo Ho, 2021] including all figures and tables for all experiments explored in this thesis. Our implementation for this collection of analysis tools and for all neural models is available from `https://github.com/ngohoanhkhoa/Generative_Probabilistic_Alignment_Models`.

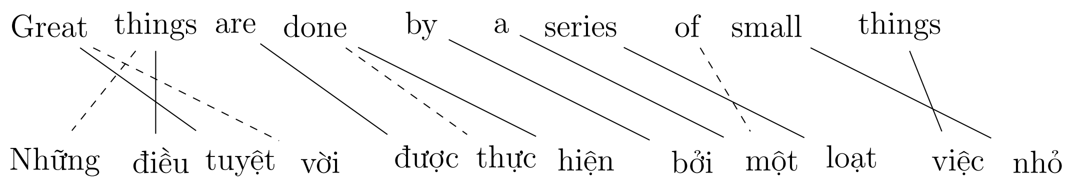


Figure 1.2: Mistakes (dashed lines) by the IBM models for the word alignment task. We can see that English word “Great” should align with both “ tuyệt” and “ vời”. In the case of asymmetrical alignment, a English source word cannot align with more than two Vietnamese target words. Another issue is that “s” in “things” should align with “ Những". This requires a alignment between smaller units.

1.1 Contributions

- We propose a collection of tools that help us to comprehensively observe all possible benefits/limitations of statistical and neural word alignment models. These tools allow us to explore in depth the main difficulties of alignment, related to aligned/unaligned words, rare/unknown words, function/content words, and word order divergences, etc. Moreover, they suggest ways to overcome these problems. We analyze the two statistical word alignment systems `Giza++` and `Fastalign` using these tools and the parallel corpora for six language pairs: English with French, German, Romanian, Czech, Japanese and Vietnamese.

- We propose neural variants for IBM style word alignment models including context-independent models, contextual models, and character-based models, which allow us to establish strong baselines for further studies. We also report a systematic comparison of these models, revealing that neuralized versions of standard alignment models vastly outperform their discrete counterparts.

- We explore variants of a fully generative neural model based on variational autoencoders to improve word representations and demonstrate that these variants can yield competitive results as compared to statistical word alignment models and to a strong neural network alignment system. Our proposed models aim to generate more symmetrical alignments. These models can be viewed as a deep learning implementation of the idea that a parallel source and target sentence should share an underlying latent representation [Melamed, 2000].

- We analyze Byte-Pair-Encoding, a subword tokenization algorithm which breaks a word into a sequence of smaller pieces. We try to identify benefits and limitations of this process for the word alignment task. Moreover, we make recommendations regarding subword configurations which help to improve word alignment performance for our six language pairs.

1.2 Thesis outline

Chapter 2 formally presents an overview of the alignment task. In this chapter, we define the generic “bitext” alignment problem at various levels from the document-level to the subword-level. We present the main models in document alignment, sentence alignment, and also sub-sentential alignment. Regarding sub-sentential alignment, we mainly discuss word alignment models under unsupervised learning and supervised learning. For such levels, various types of alignment are introduced and we report several methods to encode units for word alignment. We also present models for phrase alignment and models for structure alignment. Briefly, we would like to present the state of the art for the alignment task.

Chapter 3 presents how we efficiently evaluate alignment models. We first describe our training and test corpora for six language pairs English with French, German, Romanian, Czech, Japanese, and Vietnamese. We then explore a list of problems based on these corpora. The first issue relates the evaluation of the performance measure the performance of the unsupervised generative word alignment models. We present several common difficulties in the word alignment task: unaligned words, unknown words, alignment types, word orders, part-of-speeches, and symmetrical alignments. We perform these analyses to evaluate our baselines: two statistical word alignment tools `Giza++` and `Fastalign`. In sum, we would like to discuss limitations of the discrete models.

Chapter 4 presents an overview of neural networks and detail the most common architectures used in NLP. We then survey past attempts at using neural nets for the word alignment task. We demonstrate in this chapter the effectiveness of neural network models for the task of word alignment. Several variants of neural models that vastly outperform their discrete counterparts are proposed. We also analyze typical alignment errors of the baselines that our models overcome. In a word, we would like to illustrate the benefits and the limitations of neural networks for morphologically rich languages.

Chapter 5 discusses variational autoencoders that are useful for language generation tasks. In this chapter, we study these models for the task of word alignment, propose and assess several evolutions of a vanilla variational autoencoders. Our results confirm the previous findings about variational autoencoders and open new avenues to introduce symmetrization constraints and incorporate monolingual data. We demonstrate that these techniques can yield competitive results as compared to the statistical word alignment systems and to a strong neural network alignment system. To sum up, we introduce several models for the word alignment task.

Chapter 6 details how to perform the word alignment task by using alignment links between subwords. We explore a subword tokenization algorithm i.e., BPE and try identify its benefits and limitations for the word alignment task under different aspects such as rare words, sequence lengths and symmetrization. Note that choosing the tokenization gives an extra degree of freedom for the word alignment task. We also discuss how to select an appropriate configuration of BPE for our six language pairs. In brief, we would like to confirm if BPE is actually helpful for our task.

Chapter 7 concludes this thesis with a summary of contributions and prospects for future research.

1.3 Publications

- Anh Khoa Ngo Ho, François Yvon. Neural Baselines for Word Alignments. 16th International Workshop on Spoken Language Translation, Nov 2019, Hong-Kong, China.
- Anh Khoa Ngo Ho, François Yvon. Generative latent neural models for automatic word alignment. Proceedings of the 14th Conference of the Association for Machine Translation in the Americas (Volume 1 : Research Track), Oct 2020, USA.

Chapter 2

An overview of alignment models

In this chapter, we introduce the task of alignment for bitext at various levels from document-level to subword-level. This task aims to uncover the hidden patterns between a text in a source language and its translation in another language. We describe the generic bitext alignment problem in Section 2.1. We then discuss the three main levels of bitext alignment and also their applications in Section 2.2: document-level (Section 2.2.1), sentence-level (Section 2.2.2) and sub-sentential-level (Section 2.2.3). We present in detail word alignment, the most common level in sub-sentential alignment in Section 2.3. Generative word alignment modes `IBMs` and `HMM` are described in Section 2.4.

Contents

2.1 Bitext alignment

The term bitext refers to the parallel resources which in our study are the original documents and their translations in another language[Véronis, 2000, Melamed, 2001, Indurkhya and Damerau, 2010, Tiedemann, 2011]. Collections of bitexts also called parallel corpus, share the same domain related to a specific socio-cultural context. The text on each side could be a collection of documents, a single document, a paragraph, or a sentence. The alignment task identifies correspondences between the elements of the text in the source language and their translation in the target language. This equivalence is hierarchically structured at multiple levels: Alignments can exist between documents, paragraphs, sentences, phrases, clauses, words, and also subwords e.g. Figure 2.1. This process allows us to discover hidden patterns in the original texts and also the translated texts, which is important in many research areas such as word sense disambiguation, terminology extraction, computer-aided language learning, translation memory cleaning, and especially machine translation.

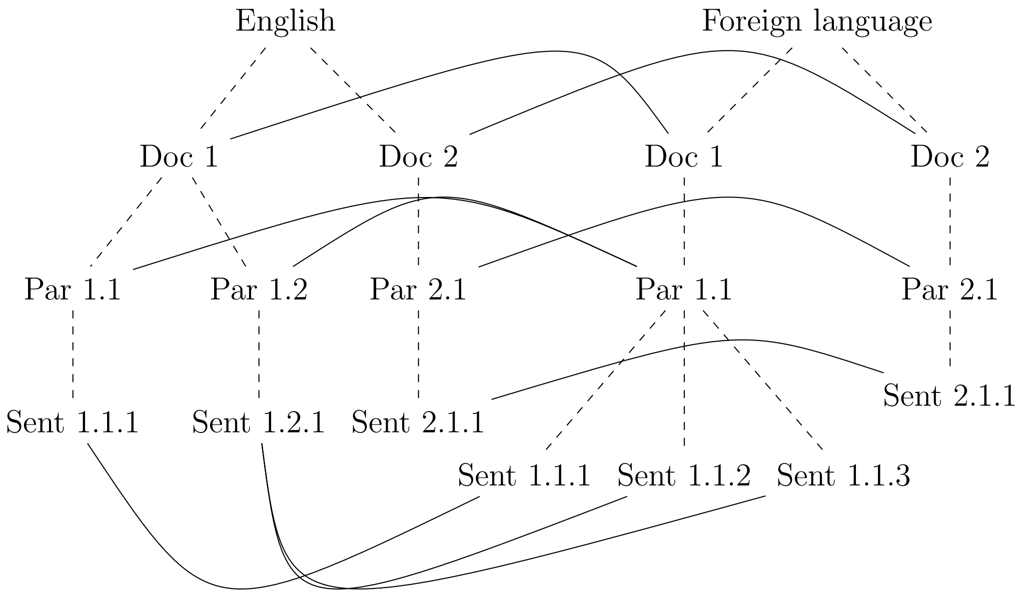


Figure 2.1: Example of an hierarchical alignment at the document (doc), paragraph (par), sentence (sent) level

2.2 Alignment granularity

The task of alignment exposes the correspondence decomposed in multiple levels from document level to character level. We discuss in this section the three main levels: document, sentence, and word alignment.

2.2.1 Document alignment

The first alignment task is to link corresponding documents with one another [Braschler and Schäuble, 1998]. This task depends mainly on the source and the meta-information available for the data collection. In some cases, this kind of mapping is provided by multilingual institutions and agencies such as the Canadian Hansard and the United Nations. Resnik [1999], Resnik and Smith [2003] propose ways to mine the web for parallel documents from multilingual websites. Extracting parallel documents from comparable corpora is also the potential approaches shown in [Steinberger et al., 2002, Pouliquen et al., 2004, Fung and Cheung, 2004a,b, Paetzold et al., 2017]. Patry and Langlais [2005] align documents across parallel corpora. Tao and Zhai [2005] propose a general method to extract comparable bilingual text without using any linguistic resources. Their method is based on an assumption that words in different languages should have similar frequency correlation if they are actually translations of each other. Vu et al. [2009] present a feature-based method to align documents with similar content across two sets of bilingual comparable corpora from daily news texts. Munteanu and Marcu [2013] use a word-by-word translation of each source document as a query to retrieve similar content target documents. A more recent project along these lines is Paracrawl[1], which aims for the development of parallel corpora for all EU languages [Esplà et al., 2019]. One of the outputs of these projects is a free/open-source pipeline. This pipeline covers fives stages from crawling data from websites on the Internet to delivering a clean parallel corpus: (a) downloading HTML

[1]Broader/Continued Web-Scale Provision of Parallel Corpora for European Languages. https://paracrawl.eu/

documents from the Internet; (b) pre-processing, normalizing and augmenting information from these documents; (c) aligning documents that are parallel; (d) aligning the segments in each of the document pairs identified; (e) filtering noisy data, deduplicating and formatting the output.

2.2.2 Sentence alignment

The next important level is the sentence level. A linguistics sentence expresses various functions based on a meaningful grammatical structure such as statement, question, exclamation, request, or command. The result of this task, called also parallel sentences, is nowadays known as the most important resource of many applications in machine translation. A sentence is not always translated into a single sentence. For instance, a long sentence could be broken up, or many short sentences could be merged. Moreover, the boundary of a sentence is hard to determine in some languages because there is no clear indication of a sentence end,e.g. Thai. Note that most of the sentence alignments are one-to-one mappings (monotonous alignment), which requires some simple constraints to obtain reasonably good alignment results. This level of alignment could be improved by the information of higher levels such as paragraphs, sections, chapters, or lower levels such as word/subword alignment.

The models of Brown et al. [1991], Gale and Church [1993] are exclusively based on sentence length. Simard et al. [1993] examine the weaknesses of Gale and Church [1993] and discuss how "cognates" would help to overcome them. In fact, for related languages, cognates provide reliable, low-cost word-level alignments, thus they can help sentence-level alignment in various ways. Cognates can be used as anchor points. Simard et al. [1993] use (word-level) cognates as an indicator of sentence alignment link quality. In addition, Chen [1993], Kay and Roscheisen [1993], Dagan et al. [1993], Utsuro et al. [1994], Wu [1994], Kueng and Su [2002] follow this line of research, discovering lexical information to improve the sentence alignment. A study of Li et al. [2010] employs a combination of both length-based and lexicon-based algorithm.

Other features are also considered in the sentence alignment algorithm such as spelling similarity, geometric and pattern recognition[2] [Melamed, 1996b, 1999]. This geometric property of the alignment map notably exploits the fact that alignment links are almost always monotonous and tend to lie near the diagonal. Singh and Husain [2005] analyze several open-source sentence alignment packages developed by Brown et al. [1991], Gale and Church [1993], Melamed [1999], Moore [2002][3]. Xu [2016] discusses some more recent models such as Hunalign [Varga et al., 2007], Gargantua [Braune and Fraser, 2010], Bleualign [Sennrich and Volk, 2011], Yasa [Lamraoui and Langlais, 2013], etc. Note that there are other alignment tools such as UPlug [4] [Tiedemann, 2003], Champollion Tool Kit (CTK) [5], Align[6]. The recent research of Schwenk [2018], using neural networks, explores a joint multilingual sentence representation and use the distance between sentences in different languages to filter noisy parallel data and to mine for parallel sentences in huge monolingual texts. Note that they do not uses any additional feature or classifier and that they apply the same approach to all language pairs. Based on this work, Artetxe and Schwenk [2019] propose the Laser which generates multilingual sentence representations for 93 languages, belonging to more than 30 different families and written in 28 different scripts. This model uses a single Bi-LSTM encoder with a shared BPE vocabulary for these languages. They also introduce a new test set of aligned sentences in 112 languages and their approach yields a strong result in multilingual similarity search even for low-resource languages.

[2]https://nlp.cs.nyu.edu/GMA/

[3]Bilingual sentence aligner (Microsoft): https://elrc-share.eu/repository/browse/bilingual-sentence-aligner/33e6526661e011e9a7e100155d026706df2f0c91489a44b78cf684b31d36d412/;
Vanilla: https://github.com/clarinsi

[4]https://github.com/Helsinki-NLP/Uplug

[5]http://champollion.sourceforge.net/

[6]http://www.cs.cmu.edu/ aberger/software/align.html

2.2.3 Sub-sentential alignment

Sub-sentential alignment is the task of exploring translational correspondences below the sentence level. It requires sentence-aligned parallel texts as its input and aims to align translational correspondences at the sub-sentential level: words, phrases clauses, and expressions. It can also rely on a bilingual dictionary to retrieve lexical correspondences.

2.2.3.1 Word alignment

We explore the most common level for sub-sentential alignment: Word alignment. The term "word" refers to a meaningful unit (token) such that a sequence of these units represents a sentence. This term could be different, depending on the language-specific definition of a word boundary. Word alignment is used to learn bilingual dictionaries, to train statistical machine translation (SMT) systems, to filter out noise from translation memories or in quality estimation applications [Specia et al., 2018].

Given a pair of sentences consisting of a sentence in a source language and its translation in a target language, word alignment aims to identify translational equivalences at the level of individual tokens [Och and Ney, 2003]. There are two main types of tasks: supervised and unsupervised learning.

Until recently, the most successful generative alignment models were statistical, as represented by the IBM Models [Brown et al., 1993b] and the HMM model Vogel et al. [1996]. These models use unsupervised estimation techniques to build asymmetrical alignment links at the word level, relying on large collections of parallel sentences. We comprehensively discuss **word alignment** in Section 2.3 and **unsupervised generative models** in Section 2.4. Melamed [2000] proposes a monolink alignment model that the noisy-channel assumption is ignored. Note that this model only considers one-to-one and null links. Cromières and Kurohashi [2009] suggest a training and a decoding procedure for this model and consider the use of syntactic trees for alignment and translation. Lardilleux et al. [2012, 2013] propose Anymalign relying on association scores between words or phrases, based on recursive binary segmentation and on document clustering. This model allows the processing of multiple languages simultaneously without any distinction between source and target. This means that this model is amenable to massive parallelism, scales easily, and is very simple to implement.

Several remarkable tools for word alignment task are Moses[7], Giza++ [Och and Ney, 2003], Fastalign [Dyer et al., 2013], Twente[8], The PLUG Word Aligner (PWA)[9], Kvec++[10], UPlug[11], SWIFT Aligner [Gilmanov et al., 2014] etc. Moreover, there are tools for alignment visualization such as Alpaco[12], Lingua-AlignmentSet[13], UMIACS Word Alignment Interface, Yawat [Germann, 2008], SWIFT Aligner, Cairo [Smith and Jahr, 2000], Hand Align[14], ILink[15], UPlug etc. A tool recently proposed is Eflomal [Östling and Tiedemann, 2016], an efficient low-memory aligner. This tool helps a phrase-based statistical machine translation to produce translations of higher quality. Östling and Tiedemann [2016] through this tool, suggest that Monte Carlo sampling should actually be the method of choice for the SMT practitioner and others interested in word alignment.

Supervised discriminative alignment models Word alignment can be viewed as a supervised structured prediction task solved with discriminative machine learning techniques which

[7]http://www.statmt.org/moses/
[8]http://taalunieversum.org/taal/terminologie/tools/software.php?id=97
[9]https://cl.lingfil.uu.se/plug/pwa/
[10]https://www.d.umn.edu/ tpederse/parallel.html
[11]https://github.com/Helsinki-NLP/Uplug
[12]https://www.d.umn.edu/ tpederse/Code/Readme.Alpaco-v0.3.txt
[13]https://metacpan.org/release/Lingua-AlignmentSet
[14]http://users.umiacs.umd.edu/ hal/HandAlign/index.html
[15]http://nlplab.org/

usually require labeled training data. These models avoid the (potentially) complicated generation process (compared to unsupervised generative models) and can accommodate rich feature sets. The simplest approach is to directly estimate, for each target word, the probability of the alternative alignment decisions which range over the source positions. This can be done using a popular multi-class classification framework called MaxEnt. Ittycheriah and Roukos [2005] propose to model the conditional alignment distribution using a log-linear model. Ayan and Dorr [2006] discuss an approach to combining outputs of existing word alignment systems. They reduce the combination problem to the level of alignment links and use a maximum entropy model to learn whether a particular alignment link is included in the final alignment. Tomeh [2012] propose a maximum entropy framework for statistical machine translation.

Another approach is to use Conditional Random Fields (CRF). This approach is explored in the discriminative sequence labeling model of Blunsom and Cohn [2006] that directly encodes the alignment distribution. The model consists of a structure similar to the HMM alignment mode and efficient learning algorithms are available through adaptations of the Viterbi and forward-backward algorithms [Getoor and Taskar, 2007]. In addition, CRFs incorporate a rich set of features, even including alignment scores of complicated generative models such as IBM 4. Note that as the HMM, the CRFs alignment model encodes asymmetrical word alignments. Therefore, we have to use standard heuristics to perform the symmetrization.

Liu et al. [2005] present a log-linear framework for symmetric word alignment. In this model, they consider three types of features: IBM 3 scores, cross-lingual POS transition scores, and dictionary-based word match scores. Their decoding step uses greedy search and they compute marginals using N-best lists. Moore [2005] consider a similar model where features strongly rely on word co-occurrence and alignment link frequencies. However, in this work, their decoding step is performed using beam-search with a modified version of an averaged perceptron. These two models operate at the alignment level and make no structural assumptions, thus they both face difficult inference problems. Niehues and Vogel [2008] show that the integrating a multitude alignment matrix can be represented by a two-dimensional CRF. They show that a multitude of features using the various knowledge sources does help to improve the performance. We also refer to Tomeh [2012] for an exhaustive presentation of supervised word alignment methods.

2.2.3.2 Phrase alignment

The phrase alignment task takes contiguous word sequences, called phrases, as translation units. In other words, a phrase alignment allows for multiple words to be grouped and linked as if they would represent a single text unit. Compared with word alignment, this task can explicitly represent a many-to-many translation relationship. A phrase pair represents an association between a source and a target phrase. For phrase alignment, the sequences of words considered are not necessarily linguistically motivated, allowing the translation of non-compositional phrases [Lin, 1999], e.g. “spass am" and “fun with the". Moreover, this task naturally captures the local context for translation. Phrase alignments can be learned in an unsupervised way without any linguistic resource, which makes the methodology generally applicable to any language pairs. This means that the more data is used in the training procedure, the longer phrases can be learned.

Phrase-based statistical machine translation systems typically require a phrase translation table, which provides a list of foreign translations and their probabilities for phrases of the original language. Such models are induced from word alignments, which means that phrase pairs are heuristically extracted from alignments between words. Koehn et al. [2003] learn phrase pairs by collecting all aligned groups of words that are consistent with word alignments generated by the Giza++ [Och and Ney, 2000]. Och and Weber [1998], Och et al. [1999], Och and Ney [2004] replace phrase pairs by alignment templates. These template describes the alignment between word classes rather than words. Venugopal et al. [2003] also extract phrase pairs from word alignment models by leveraging the maximum approximation as well as the word lexicon. In addition, there is a work of Wisniewski et al. [2010] explores a methodology for analyzing the errors of a phrase-based translation system.

Phrase translation models can be learned directly from phrase alignment models. Marcu and Wong [2002] propose the joint phrase model with a generative story: (a) creating a number of concepts; (b) generating a foreign and English phrase from each concept; (c) reordering the English phrases. This concept can be considered as an abstraction of phrase types. The model jointly generates both foreign and English words from a concept, which explicit phrase alignments. An analysis for this is in [DeNero et al., 2008]. Zhang and Vogel [2005] describe a model efficiently processing arbitrarily long phrases because they capture more contexts than short phrases and result in better translation qualities. They demonstrate that their model is efficient in both time and space, yielding better translations.

A phrase can contain gaps and overlap arbitrarily or in some nested structure. Yamamoto et al. [2003] use sequential pattern mining algorithms from parallel strings through co-occurrence analysis, which uniformly generates both rigid and gapped sequences simultaneously. Tambouratzis et al. [2011] introduces a phrase-alignment approach involving the processing of a small bilingual corpus in order to extract suitable structural information[16]. Pal et al. [2011], Tomeh [2012] propose a framework using the information of multiword expressions to boost the performance of phrase-based SMT. In [Junczys-Dowmunt, 2012], they develop a method for the compression of the word-aligned target language in phrase tables. Cuong and Sima'an [2014] develop a phrase-based model directly trained on mix-of-domain corpora. In order to reduce the size of a phrase translation table, Nishino et al. [2016] propose an effective approach that removes the least useful phrase pairs from this table. A recent study of Bogoychev and Hoang [2016] presents a new standalone phrase table, optimized for query speed and memory locality.

2.2.3.3 Structure alignment

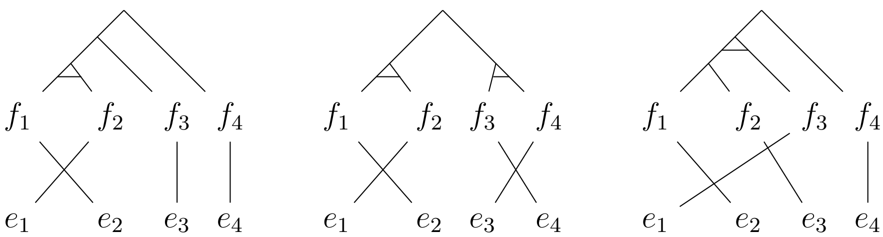


Figure 2.2: Several matchings of length four with ITG parses [Wu, 1997].

Structure alignment provides a matching between grammatical components of a sentence pair. It requires a compositional analysis for the sentences which creates segments. The purpose of this task is to build parallel treebanks, corpora including mappings between linguistically motivated analyses across languages. It is clear that any tree alignment approach is also a natural way of generating phrase correspondences. These treebanks hence can be used in cross-linguistic research [Cyrus, 2006, Rios et al., 2009], bilingual transfer rule induction [Lavoie et al., 2001, Buch-Kromann, 2007, Graham and van Genabith, 2009] and especially machine translation. Structure alignment assumes that a structure over a sentence can be decomposed into smaller units with relations between them, yielding constituents or substructures. They refer to either single tokens or several tokens from a sentence. In this scenario, constituents can overlap.

Tree alignment is a special case of structural alignment. In this case, a tree alignment is strictly compositional and hierarchical [Indurkhya and Damerau, 2010]. In other words, segments within two linked sub-trees align only with each other and there is a root segment spanning the entire sentence. Constituents within a tree are called nodes with one special node at the root of the tree. Labeled constituents are called non-terminals and single tokens are referred to as terminal nodes. Edges connecting these nodes can be also labeled. Note that

[16]The PRESEMT (Pattern REcognition-based Statistically Enhanced MT), http://presemt.eu/

word alignment needs to only be fed a sentence pair whereas a tree alignment also needs the structural annotation of this sentence. We can consider this type of alignment as a phrase alignment using additional structural constraints.

These structural constraints help to control the overlap between segments. A number of algorithms for this structure alignment consider one fixed disjoint segmentation of each monolingual sentence. This means that the segments in this segmentation do not overlap and cover the whole sentence. We can join neighbor disjoint phrases to form a tree, which helps to enrich authorized segments. Note that we can use monolingual syntactic parsers to obtain a grammatical tree, which implies that each segment represents a grammatical phrase. For word alignment, a disjoint fixed segmentation is implied while a tree alignment considers structural constraints on both sides. It should be mentioned that alignment constraints are applied to the set of links between authorized segments. To sum up, the task of alignment is to link tree nodes from one source sentence to corresponding units in the target sentence. This is based on an assumption that there is a similar structure in the target sentence.

Tree alignment (Figure 2.2) requires that both sides of the parallel corpus are analyzed syntactically. These analyses are based on entirely automatic annotation using monolingual hand-crafted or statistical parsers. This yields a problem of consistency between independent syntactic analyses where it is difficult to find a common representation describing a complete mapping from one tree to another. Therefore, generative tree alignment models are not very successful because they are based on the strong constraints given by the monolingual parses. This is why most approaches apply heuristic or discriminative models for this task of alignment.

The approach of Wu [1997] considers the crossing constraint for lexical mappings: Aligning two subtrees means that words in the yield of the first can be aligned only to words in the yield of the second. Several benefits are (a) the crossing constraint greatly reduces the space of possible alignments and thereby reduces the search complexity; (b) this constraint is accurate most of the time thanks to its relation to syntax.; (c) large-distance reordering can easily be modeled while avoiding the complexity of arbitrary permutations. Other algorithms are used to search the best alignment such as greedy top-down search algorithm [Matsumoto et al., 1993], bottom-up beam search algorithm [Grishman, 1999] and greedy best-first alignment strategies [Menezes and Richardson, 2001, Groves et al., 2004]. The approach of [Tinsley et al., 2007, Zhechev and Way, 2008] allows minor corrections in case of blocking links using various search heuristics. Lavie et al. [2008] propose an approach where alignment decisions are greedily propagated from leaf nodes to the root. Another study about the relationship between alignments and monolingual structures of sentences is dicussed in Cromières [2010].

There exist two alternatives for crossing constraint: (a) separately parsing each sentence using two distinct Context-Free Grammars (CFG) with parse-match strategy. (b) simultaneously parsing both of the sentences using a synchronous CFG, producing parses for both sides along with the alignment. Indurkhya and Damerau [2010] discuss the lack of appropriate, robust, and monolingual grammars of the former approach. It also suffers a mismatch of the grammars across languages and inaccurate selection between multiple possible constituent matchings. The major disadvantage of the latter alternative is the difficulty of obtaining the grammar.

Inversion transduction grammars (ITGs), introduced by Wu [1995, 1997], aim at a symmetric generative explanation of translated texts. The generation of sentence pairs is based on a common structure with permutations in one language allowed. In other words, ITG is a special case of syntax-directed transduction of a context-free language. It is equivalent to binary or ternary syntax-directed transduction whose rules are restricted to straight and inverted permutations only. ITG can be used to induce symmetric word alignments [Saers and Wu, 2009] and to restrict the search space of other alignment models [Cherry and Lin, 2006].

2.3 Word alignment

A word alignment is a mapping between two parallel sentences ($\mathbf{f}$, $\mathbf{e}$). The source sentence $\mathbf{f}$ consists of a sequence of J tokens $(f_1, \ldots, f_J)$ and the target sentence $\mathbf{e}$ consists of I tokens $(e_1, \ldots, e_I)$. The mapping corresponds to the set of individual links between the source and the target word positions. The word alignment is thus defined as:

$$A = \{(j, i) : 1 < j < J, 1 < i < I\} \tag{2.1}$$

Figure 2.3 displays an example of a word alignment between f_1^7 and e_1^8: $A = \{(1, 1)$, (2,2), (2,3), (3,4), (4,4), (5,5), (5,6), (6,5), (6,6), (7,7) }.

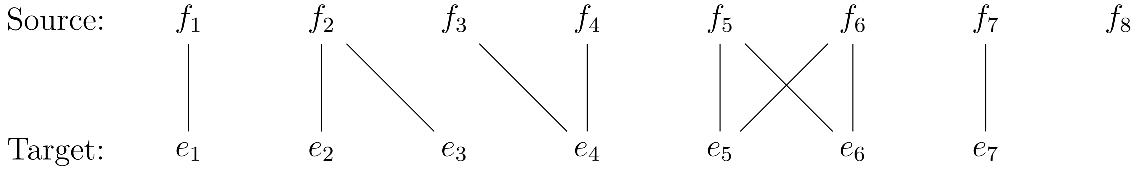


Figure 2.3: Example of a word alignment between f_1^7 and e_1^8: $A = \{(1, 1)$, (2,2), (2,3), (3,4), (4,4), (5,5), (5,6), (6,5), (6,6), (7,7) }

2.3.1 Different types of mapping

Each language is characterized by specific compounding, agglutinative and morphological features, yielding various manners to express a concept. This means that the association between concepts sometimes yields associations that go beyond the direct association between one source and one target word, and take the form of more complex link patterns such as one-to-many, many-to-one, many-to-many links or even null links. These are illustrated below.

One to one alignments English word "understandable" is translated by one French word "compréhensible", which gives a one-to-one link (Figure 2.4). Therefore, a one-to-one alignment is such that one source word and one target word are only aligned together. In other words, these source and target word positions appear in exactly one link. This is the case of links (1,1) and (7,7) in Figure 2.3.

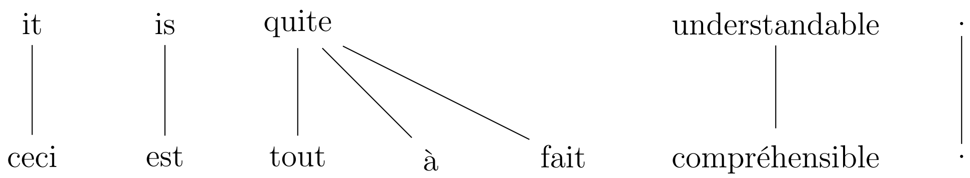


Figure 2.4: Example of a word alignment: One to one alignments (("it", "ce"), ("is", "est"), ("understandable", "compréhensible"), (".",".")) and one to many alignments (("quite", "tout"),("quite", "à"),("quite", "fait"))

One to many/ many to one alignments One-to-many mapping are such that a source word is linked to more than one target words, e.g., one to many links (2,2), (2,3) in Figure 2.3. Many-to-one mapping is the reverse case, where a target word is linked to more than two source words, e.g. the links (3,4), (4,4) in Figure 2.3. Another example is the French multi-word expression "tout à fait" which is often translated as one single English word "quite" (Figure 2.4). Moreover, the corresponding units are not necessarily contiguous, when the source or the target sentence contains a flexible multiword expression, e.g. a separable or phrasal verb (Figure 2.5).

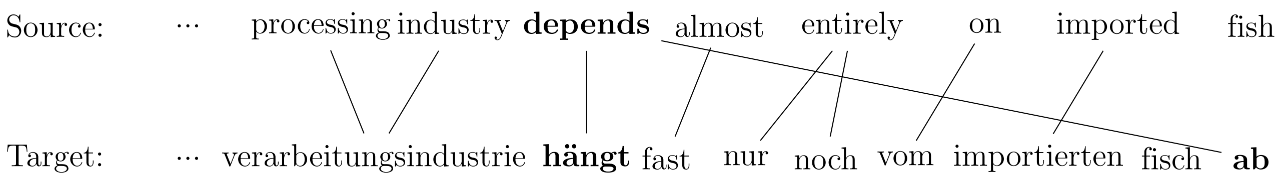


Figure 2.5: Example of discontiguous correspondences: English word "depends" aligns with two German words "hängt" and "ab".

Many to many alignments The final case is many to many links, corresponding to the situation where more than two source words and more than two target words are aligned together, e.g. many to many links (5,5), (5,6), (6,5), (6,6) in Figure 2.3. The links (5,6) and (6,5) are also called crossing links. We observe this type of alignment in a sentence pair such as ("The poor don't have any money", "Les pauvres sont démunis") where the English words "don't", "have", "any", "money" are linked to the French words "sont" and "démunis" (Figure 2.6).

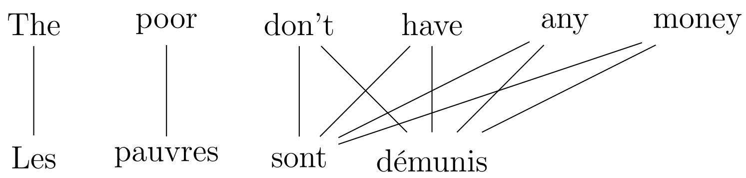


Figure 2.6: Example of a word alignment: the English words "don't", "have", "any", "money" are linked to the French words "sont" and "démunis".

Unaligned word and null link Word f_8 in Figure 2.3 is unaligned and is not linked to any target word. Asymmetrical alignment models such as IBM Models and HMMs (Section 2.4) apply the functional constraint that every source words is linked to exactly one target word. This constraint only licences one-to-one and many-to-one mappings. Therefore, the word f_8 is linked to a special NULL token on the target side (Figure 2.7). This link is called a null link.

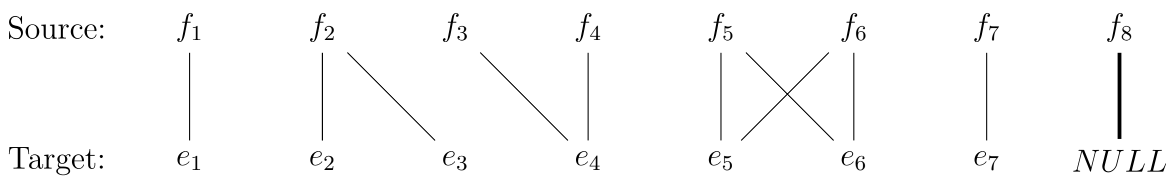


Figure 2.7: Example of a null link: $(f_8, NULL)$

2.3.2 Encoding units for word alignment

In our research, we evaluate our models based on word-level alignment. Besides the information from words, we expect that considering smaller units such as byte pair encoding (BPE) could improve the performance of our models. For agglutinative languages such as Turkic languages, this helps to produce finer-grained alignments, i.e. alignments between morphemes or language features. Moreover, another benefit of this subword tokenizations is to handle large and open vocabulary, specially reducing the problem of rare words.

Byte pair encoding Byte pair encoding is a form of data compression introduced by Gage [1994]. BPE subword tokenization[17] breaks a word into a sequence of smaller pieces, yielding

[17] Tools for BPE tokenization: https://github.com/rsennrich/subword-nmt, https://github.com/google/sentencepiece

rare words to be split up into more frequent subwords. For instance, a French word "yaourter" could be broken into "yaourt" and "er". The BPE algorithm consists of three steps: (a) The algorithm starts with a vocabulary of characters. (b) It then iteratively selects the most frequent n-gram pairs to be included in the unit inventory. (c) The algorithm stops when it reaches the desired vocabulary size.

The BPE algorithm determines the vocabulary size by controlling the balance between character level and word level tokenization. This is also an approach for morphologically rich languages, where the root word is exposed, e.g. "act" in the words "act-or", "act-ing", "re-en-act". A BPE sequence is always longer than the corresponding sequence of words, leading to a more complex alignment with a larger number of links. BPE is used in many machine translation models [Sennrich et al., 2016, Morishita et al., 2018, Shapiro and Duh, 2018, Wang et al., 2020, Garg et al., 2019, Liu et al., 2019]. Note that subwords can be generated by using morpheme segmentation [Nießen and Ney, 2000, Luong et al., 2013] and unigram language models [Kudo and Richardson, 2018] besides BPE. A recent segmentation algorithm called Dynamic Programming Encoding (DPE) is proposed in He et al. [2020]. They use a lightweight mixed character-subword transformer as a means of pre-processing parallel data to segment sentences. Ding et al. [2019a] makes recommendations regarding the selection of proper BPE configurations by comparing different NMT architecture and reporting BLEU scores.

Our results and analyses are based on word-level alignments. Subword-level alignments are converted into word-level alignments as follows: a link between a source and a target word exists if there is at least one link alignment between their subwords [Garg et al., 2019]. Note that BPE could serve two purposes: (a) Train representations for unknown words. (b) BPE alignments are used in word alignments. Figure 2.8 displays an example of the conversion from subword-level to word-level alignment.

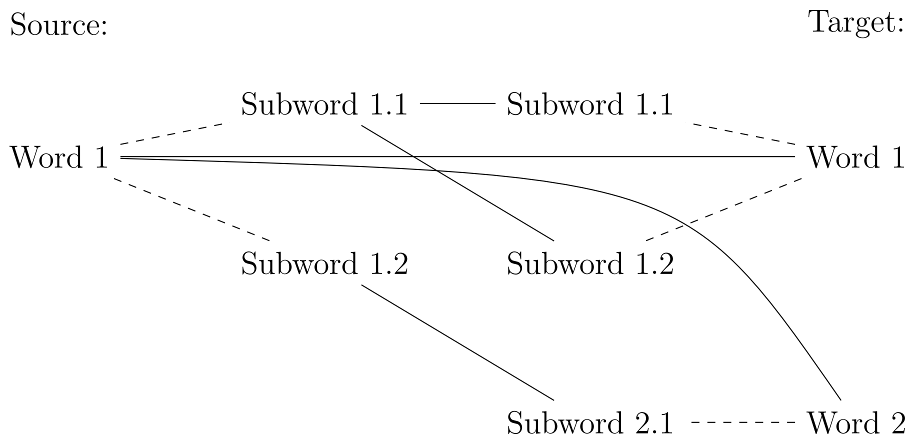


Figure 2.8: Example of a subword alignment: The subword-level links (1,1), (1,2), (2,3) become the links (1,1), (1,2) in the word level alignment

2.4 Unsupervised generative alignment models

Let's first recall the definition of a word alignment. A word alignment is a mapping between two parallel sentences (f_1^J, e_1^I). The source sentence f_1^J consists of a sequence of J tokens $(f_1, \ldots, f_J)$ and the target sentence e_1^I consists of I tokens $(e_1, \ldots, e_I)$. The word alignment is defined as: $A = \{(j,i) : 1 < j < J, 1 < i < I\}$.

In statistical machine translation [Brown et al., 1993b], they model the translation probability $P(f_1^J|e_1^I)$, which describes the relationship between a source sentence f_1^J and a target sentence e_1^I. They add latent alignment variables $a_1^J = (a_1, \ldots, a_J)$ with $a_j \in [0 \ldots I]$ to the translation model. Therefore, they obtain an asymmetric alignment model associating each word in a source sentence f_1^J with exactly one word from the target sentence $e_0^I = e_0 \ldots e_I$

of $I+1$ words [Och and Ney, 2003]. The target sentence is completed with a NULL symbol, conventionally at index 0. $P(f_1^J|e_1^I)$ can be modeled as:

$$P(f_1^J|e_1^I) = \sum_{a_1^J} P(f_1^J, a_1^J|e_1^I) \tag{2.2}$$

The probabilistic model is thus decomposed as:

$$P(f_1^J, a_1^J|e_1^I) = P(J|e_1^I) \prod_{j=1}^{J} P(f_j|f_1^{j-1}, a_1^j, e_1^I)\, P(a_j|f_1^{j-1}, a_1^{j-1}, e_1^I) \tag{2.3}$$

where $p(J|e_1^I)$ is the probability predicting the number of words in the source sentence given the target sentence. The two terms in the inner product in equation (2.3) are referred to respectively as the lexical probability (lexical model) and the link probability (distortion model).

The Viterbi alignment $\hat{a}$ given a sentence pair (f_1^J, e_1^I) is defined as:

$$\hat{a}_1^J = \text{argmax}_{a_1^J} P(f_1^J, a_1^J|e_1^I) \tag{2.4}$$

2.4.1 Unsupervised learning: Expectation Maximization

Given a corpus of parallel sentences $\{\mathbf{f}_k, \mathbf{e}_k\}_1^K$ including K sentence pairs and the alignment variable denoted as $\mathbf{a} = a_1^J$, we can estimate the parameters θ of the model $P_\theta(\mathbf{f}|\mathbf{a}, \mathbf{e})$ without any alignment information. We assume that all sentence pairs are independent and identically distributed and they represent sufficiently the entire population of translated sentences. As the alignment variable $\mathbf{a}$ is not observed, the objective of maximum likelihood estimation for an incomplete training set is defined as:

$$\hat{\theta} = \text{argmax}_\theta \sum_{k=1}^{K} \log \sum_{\mathbf{a}} P_\theta(\mathbf{f}_k, \mathbf{a}|\mathbf{e}_k) \tag{2.5}$$

For this optimization, one of the techniques well used is Expectation-Maximization (EM), an iterative re-estimation algorithm [Dempster et al., 1977]. This algorithm adjusts the model parameters step by step by improving the likelihood of observable data. The main idea is to fill the gaps of the incomplete data i.e., alignment variable $\mathbf{a}$ with the expected values according to the current model. Note that EM is theoretically guaranteed to never decrease the data likelihood in any iteration, however it could be stuck in a local maximum.

Another technique is Gibbs sampling [Gelfand and Smith, 1991], a special case of the Markov Chain Monte Carlo (MCMC) method, used in Eflomal [Östling and Tiedemann, 2016]. The idea in Gibbs sampling [Lynch, 2007] is to generate posterior samples by sweeping through each variable to sample from its conditional distribution with the remaining variables fixed to their current values. We can summarize Gibbs sampling in two steps: (a) Derive the full joint density and the posterior conditionals for each of the random variables in the model. (b) Simulate samples from the posterior joint distribution based on the posterior conditionals.

Expectation-Maximization EM starts with an arbitrary initial parameter θ_0, iterates between computing the posterior probabilities of individual alignments $\{\mathbf{a}_k\}_1^K$ for the entire corpus and updating the parameters θ.

- Expectation (E-step): Given the parameters θ_t at the time step t, the algorithm computes the posterior $q_{\theta_t}(\mathbf{a}_k)$ for each sentence pair $(\mathbf{f}_k, \mathbf{e}_k)$:

$$q_{\theta_t,k}(\mathbf{a}_k) = p_{\theta_t}(\mathbf{a}_k|\mathbf{f}_k, \mathbf{e}_k) = \frac{p_{\theta_t}(\mathbf{f}_k, \mathbf{a}_k|\mathbf{e}_k)}{p_{\theta_t}(\mathbf{f}_k|\mathbf{e}_k)} \tag{2.6}$$

- Maximization (M-step): Considering all alignments $\{\mathbf{a}_k\}_1^K$ at the time step t, the new parameters θ_{t+1} are estimated as:

$$\theta_{t+1} = \text{argmax}_\theta \sum_{k=1}^{K} \sum_{\mathbf{a}} q_{\theta_t,k}(\mathbf{a}) \log p_\theta(\mathbf{f}_k, \mathbf{a}|\mathbf{e}_k) \tag{2.7}$$

2.4.2 IBM models and derivative alignment models

We describe the many-to-one alignment models which associate each source word with exactly one word from the target sentence: the IBM models proposed by Brown et al. [1993b] and the HMM-based model of Vogel et al. [1996] constitute the foundation of studies on word alignment. Several highly-optimized implementations of these models are widely used in NLP research practices, such as `Giza++` [18] [Och and Ney, 2003] and `Fastalign` [19] [Dyer et al., 2013]. In our evaluation and analysis, we observe the results of these tools (see Chapter 3). Note that these probabilistic models use the Expectation-Maximization (EM) algorithm to adjust model parameters.

2.4.2.1 IBM Model 1 (`IBM-1`)

`IBM-1` is the simplest model with the strongest independence assumptions. $p(J|e_1^I)$ is simplified as $p(J|I)$. The lexical probability depends only on aligned target words, which means that the dependency on all previous words and previous alignment links is ignored $p(f_j|f_1^{j-1}, a_1^j, e_1^I) = p(f_j|e_{a_j})$. The distortion model is a uniform distribution $p(a_j|f_1^{j-1}, a_1^{j-1}, e_1^I) = \frac{1}{(I+1)}$. `IBM-1` is thus based on the lexical model. Therefore, the joint distribution $p(f_1^J, a_1^J|e_1^I)$ is rewritten as:

$$p(f_1^J, a_1^J|e_1^I) = \frac{p(J|I)}{(I+1)^J} \prod_{j=1}^{J} p(f_j|e_{a_j}) \tag{2.8}$$

The parameters of this model are $\theta = p(f|e), \forall(f,e) \in V_f \times V_e$ where V_f and V_e are respectively source and target vocabulary with fixed size depending on the training corpus. Note that V_e includes also the NULL word. The number of parameters hence is $|V_f| \times |V_e|$. `IBM-1` guarantees that the global optimum is always found because the objective function is convex [Brown et al., 1993b]. However, Toutanova and Galley [2011], Simion et al. [2015] point out that IBM-1 is not strictly convex, the same optimum could be achieved by the different sets of parameters. This highlights the importance of the parameter initialization in practice.

Another important property of `IBM-1` is that the inference procedure can be performed exactly and that the optimal alignment can be computed on a per position basis. Alignment decisions are made completely independently from one another, based on word co-occurrence. The two words co-occurring frequently do not mean that they should be linked, which is called indirect associations [Melamed, 2000]. Consider, for example, the word "the" in English and "et" in French. Both are very frequent and their high co-occurrence rate is accidental and does not imply that they should be aligned. Another nice example is proper names: Moby Dick co-occurs with Moby Dick, this does not mean that Moby (French) should align with Dick (English). Moreover, it is impossible to control the number of source words aligned to some target words due to the lack of distortion model. Moore [2004] adds one more limitation that `IBM-1` has only one NULL token. These issues are taken into account in more complex models such as `IBM-2` or `HMM`.

[18] http://www.statmt.org/moses/giza/GIZA++.html
[19] http://github.com/clab/fastalign

2.4.2.2 IBM Model 2 and its reparameterization - `Fastalign`

A new assumption about the dependency on absolute token positions is introduced in this model, providing a richer distortion model $p(a_j|f_1^{j-1}, a_1^{j-1}, e_1^I) = p(a_j|j, I, J)$:

$$p(f_1^J, a_1^J|e_1^I) = p(J|I)\prod_{j=1}^{J} p(f_j|e_{a_j})p(a_j|j, I, J) \tag{2.9}$$

The dependency on J is usually ignored to reduce the number of parameters $p(a_j|j, I)$. This model includes two separate components that can be understood as processing lexical translation and then reordering the words. This helps to produce a different score of the likelihood for each alignment pattern due to position parameters. However, achieving a global maximum with EM is not guaranteed anymore since the likelihood objective is no longer concave. Because of the similarity between `IBM-1` and `IBM-2`, the lexical parameters of `IBM-2` are often initialized by the pre-trained parameters obtained from `IBM-1`.

In our work, we use the implementation provided in `Fastalign` [Dyer et al., 2013], which relies on a log-linear reparameterization of the distortion model of `IBM-2`.

$$h(i, j, I, J) = -\left|\frac{i}{I} - \frac{j}{J}\right| \tag{2.10}$$

$$p(a_j|j, I, J) = \frac{\exp(\lambda h(i, j, I, J))}{Z(j, I, J)} \tag{2.11}$$

where the resulting partition function (Z) must sum over a very large space, and approximations are often required; the value of λ controls the level of encouragement of alignment links around the diagonal.

2.4.2.3 Hidden Markov Model `HMM`

The model `HMM` assumes first-order dependencies between adjacent links [Vogel et al., 1996].

$$p(f_1^J, a_1^J|e_1^I) = p(J|I)\prod_{j=1}^{J} p(f_j|e_{a_j})p(a_j|a_{j-1}, I) \tag{2.12}$$

This model also assumes that the distortion probability $p(a_j|a_{j-1})$ or $p(i|i', I)$ only depends on the jump width $(i - i')$, which means the independence on the absolute word positions. The model uses a set of non-negative parameters $\{c(i - i')\}$, yielding the distortion probability:

$$p(i|i', I) = \frac{c(i - i')}{\sum_{i''=1}^{I} c(i'' - i')} \tag{2.13}$$

Och and Ney [2003] propose to refine the modeling of NULL words by extending the HMM network with I NULL words e_{I+1}^{2I} (instead of just one). Each target word e_i has a corresponding NULL word e_{i+I}, which helps the model to remember the previously visited target word after jumping to the NULL token. They also introduce the parameter p_0 which is the probability of a transition to the NULL word. The transitions involving NULL words in `HMM` follow the constraints:

$$p(i + I|i', I) = p_0 \tag{2.14}$$

$$p(i + I|i' + I, I) = p_0 \tag{2.15}$$

$$p(i|i' + I, I) = p(i|i', I) \tag{2.16}$$

Liang et al. [2006] uses the distortion $c(\cdot)$ with a multinomial distribution over $2N+1$ offset buckets $c(\leq -N), c(-N+1), \ldots, c(N-1), c(\geq N)$. The structure of the `HMM` provides an adequate basis for many extensions e.g., the research of Toutanova et al. [2002] with boosting lexical translation probabilities using part-of-speech tags, building the better null alignments, and incorporating the notion of fertility; Schulz et al. [2016] with the non-null model; Deng and Byrne [2006] with word-to-phrase alignment models and models with included morphology [Burlot and Yvon, 2017]. In our works, we apply neural networks into the lexical model and the distortion model of `HMM`.

Note that `HMM` reuses the same lexicon model as `IBM-1` and `IBM-2`. The initialization from the pre-trained parameters, in this case, is helpful because the log-likelihood function in `HMM` is not concave. The best alignment can be found using the Viterbi algorithm [Viterbi, 1967] and the expectation step in EM is efficiently done by the Baum-Welch algorithm.

Viterbi algorithm The Viterbi algorithm is a dynamic programming algorithm that computes the most probable state sequence in a `HMM`, corresponding here to a sequence of target word positions. The probability of the most probable path ending in the target word e_i with the source word f_j is expressed in the following recursive formula:

$$p_{e_i}(f_j, j) = p(f_j|e_i) \max_{e_{i'}}(p_e(f_{j-1}, j-1)p(i|i')) \tag{2.17}$$

We can thus compute recursively (from the first to the last element of our sequence) the probability of the most probable path. This algorithm is an efficient way to make an inference, or prediction, to the sequence of target word positions given the model parameters $p(f_j|e_i)$ and $p(i|i')$.

Baum-Welch algorithm (BW) The Baum-Welch algorithm is a dynamic programming approach for EM using the forward-backward algorithm. Its purpose is to compute the expectations for the state transition matrix (the distortion probabilities in our case) and the emission matrix (or the lexicon probabilities). There are a few phases for this algorithm, including the initial phase, the forward phase, the backward phase, and the update phase. The forward and the backward phase form the E-step of the EM algorithm, while the update phase itself is the M-step.

- Forward phase: $\alpha_i(j)$ is the cumulated probability of seeing the source words $[f_1, \dots, f_j]$ and being in the target word e_i at the source word f_j. π is the initial state distribution. The recursion formula for the α step is:

$$\alpha_i(1) = \pi_i p(f_1|e_i) \tag{2.18}$$

$$\alpha_i(j+1) = p(f_{j+1}|e_i) \sum_{i'} \alpha_{i'}(j)p(i|i') \tag{2.19}$$

- Backward phase: $\beta i(j)$ is the probability ending the partial sequence $[f_{j+1}, \dots, f_J]$ given starting target word e_i at source word f_j. The recursion formula for the β step are:

$$\beta_i(J) = 1 \tag{2.20}$$

$$\beta_i(j+1) = \sum_{i'} \beta_{i'}(j+1)p(i'|i)p(f_{j+1}|e_i) \tag{2.21}$$

- Update phase: The parameters of the HMM can be updated by using the posteriors of the alignment variables.

$$q(f_j|e_i) = \frac{\alpha_i(j)\beta_i(j)}{\sum_{i'} \alpha_{i'}(j)\beta_{i'}(j)} \tag{2.22}$$

$$q(i|i') = \frac{\alpha_i(j)p(i''|i)\beta_{i'}(j+1)p(f_{j+1}|e_{i'})}{\sum_{i''}\sum_{i'''} \alpha_{i''}(j)p(i'''|i'')\beta_{i'''}(j+1)p(f_{j+1}|e_{i'''})} \tag{2.23}$$

2.4.2.4 Fertility model in IBM model 3 and beyond

We briefly describe the fertility model used in IBM models 3, 4 and 5 which have a significantly more complicated structure than the simple Models 1 and 2. This fertility model learns to capture the phenomena that some target words tend to align with multiple source words while others tend to align with only one or zero words. The model introduces ϕ_i being the number of aligned source words for the target word e_i. Figure 2.9 illustrates the fertility of the English word "quite" when it translates to "tout à fait" in French. The fertility of "quite" is 3, which means that the model needs to generate three alignment links for this English word. Moreover, this also provides an alternative method of modeling null links, corresponding to $\phi = 0$, which helps to determine the unaligned words. Brown et al. [1993b] defines this fertility distribution as a function of the sentence length and introduces a parameter p_0 representing the a prior probability of a null alignment.

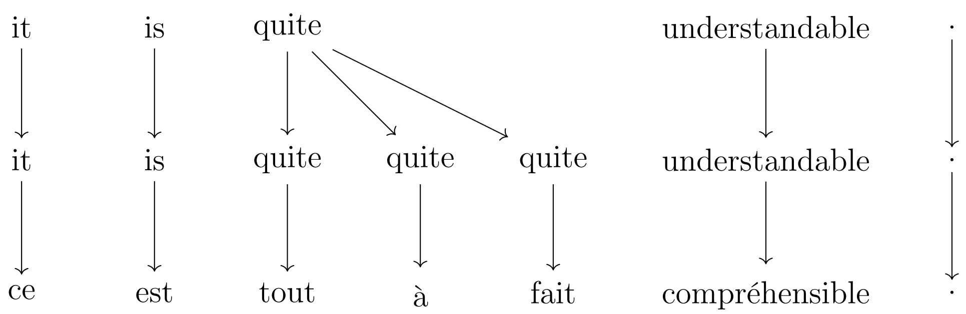


Figure 2.9: Example of fertility of the English word "quite". Note that all the other words also have a fertility (equal to 1).

The model `IBM-3` tries to integrate many remarkable properties observed in alignments, it still makes a lot of assumptions such as the independence between surrounding contexts and interaction between alignment decisions. `IBM-4` is an updated version where:

- Distortion parameters are based on a relative position, which encourages a better generalization and reduces the effect of data sparsity.
- A first-order dependency is introduced, which captures the interaction between links. This assumption is similar to the distortion component of the `HMM` model.
- Lexical information based on word classes contributes to the distortion model of `IBM-4`.

2.4.3 Symmetrization

While alignment is seemingly a symmetrical task, the probabilistic models presented above are asymmetrical in essence. A number of attempts have tried to generate symmetrical alignments, either as a built-in property of the model or as a heuristic post-processing step [Och and Ney, 2003, Koehn and Hoang, 2007].

2.4.3.1 Intersection, union and grow-diag-final

Suppose that we have two alignments with two opposite directions a_1^J and b_1^I for each sentence pair. We need to post-process heuristically the two alignments by merging them to produce a symmetric alignment. Let $A = \{(a_j, j) | a_j > 0\}$ and $B = \{(i, b_i) | b_i > 0\}$ denote the sets of alignments in the two Viterbi alignments. Various procedures have been proposed to combine A and B into one alignment matrix C:

- Intersection: $C = A \cap B$. This helps to focus on links for which both alignment models agree on, increasing precision and reducing recall. The resulting alignment only includes one-to-one links, which may hurt the precision when measured in terms of bisegment correspondences. This procedure is illustrated in Figure 2.10.

- Union: $U = A \cup B$. The union shows an opposite effect, a higher recall and a lower precision. One issue with this method is that it increases the number of garbage links. This procedure is shown in Figure 2.10.

- Grow-diag-final: The result of intersection C, which is assumed to be most reliable, is extended by adding neighbor (i, j) from the union set. The extension follows the rules:
  - The alignment (i, j) has a horizontal neighbor $(i-1, j)$, $(i+1, j)$ or a vertical neighbor $(i, j-1)$, $(i, j+1)$ that is already in C.
  - The set $C \cup \{(i, j)\}$ does not contain alignments with both horizontal and vertical neighbors.
  - The words e_i and f_j have not been linked yet.

  The growing heuristic can be different, depending on the definition of link neighbor and also the balance between the precision and the recall [Och et al., 1999, Och and Ney, 2000].

The method has proven its usefulness in phrase-based SMT [Koehn et al., 2003, Ayan and Dorr, 2006]. In our work, we use the grow-diag-final algorithm of Moses [20].

2.4.3.2 Agreement constraints

Liang et al. [2006] explore methods for incorporating constraints in HMM-based alignment training, maximizing a combination of the data likelihood and a measure of agreement between specific probability score given by the two asymmetrical models. They evaluate the agreement between $p_{\theta_1}(\mathbf{a}|\mathbf{f}, \mathbf{e})$ and $p_{\theta_2}(\mathbf{a}|\mathbf{f}, \mathbf{e})$ by summing over all alignment probabilities on which both models agree, yielding the objective function:

$$\max_{\theta_1, \theta_2} \sum_{\mathbf{f}, \mathbf{e}} [\log p_{\theta_1}(\mathbf{f}, \mathbf{e}) + \log p_{\theta_2}(\mathbf{f}, \mathbf{e}) + \log \sum_{\mathbf{a}} p_{\theta_1}(\mathbf{a}|\mathbf{f}, \mathbf{e}) p_{\theta_2}(\mathbf{a}|\mathbf{f}, \mathbf{e})] \tag{2.24}$$

E-step of EM requires to sum over the set of alignments with exclusively one-to-one mappings, which is intractable. Therefore, Liang et al. [2006] propose a simple approximation using the posterior marginal probability of individual links $p(a_{i,j}|\mathbf{f}, \mathbf{e})$. These probabilities, which are called state occupation probabilities are computed efficiently by using Baum-Welch algorithm for HMM [Matusov et al., 2004] (Section 2.4.2.3). One drawback of training this kind of model is that it is not clear what objective the approximate procedure actually optimizes. Moreover, enforcing agreement in joint training faces a problem that the two models are restricted to one-to-one alignments [Liang et al., 2006]. Liu et al. [2015] replace the original probability of

[20] The default heuristic grow-diag-final starts with the intersection of the two alignments and then adds additional alignment points. http://www.statmt.org/moses/?n=FactoredTraining.AlignWords

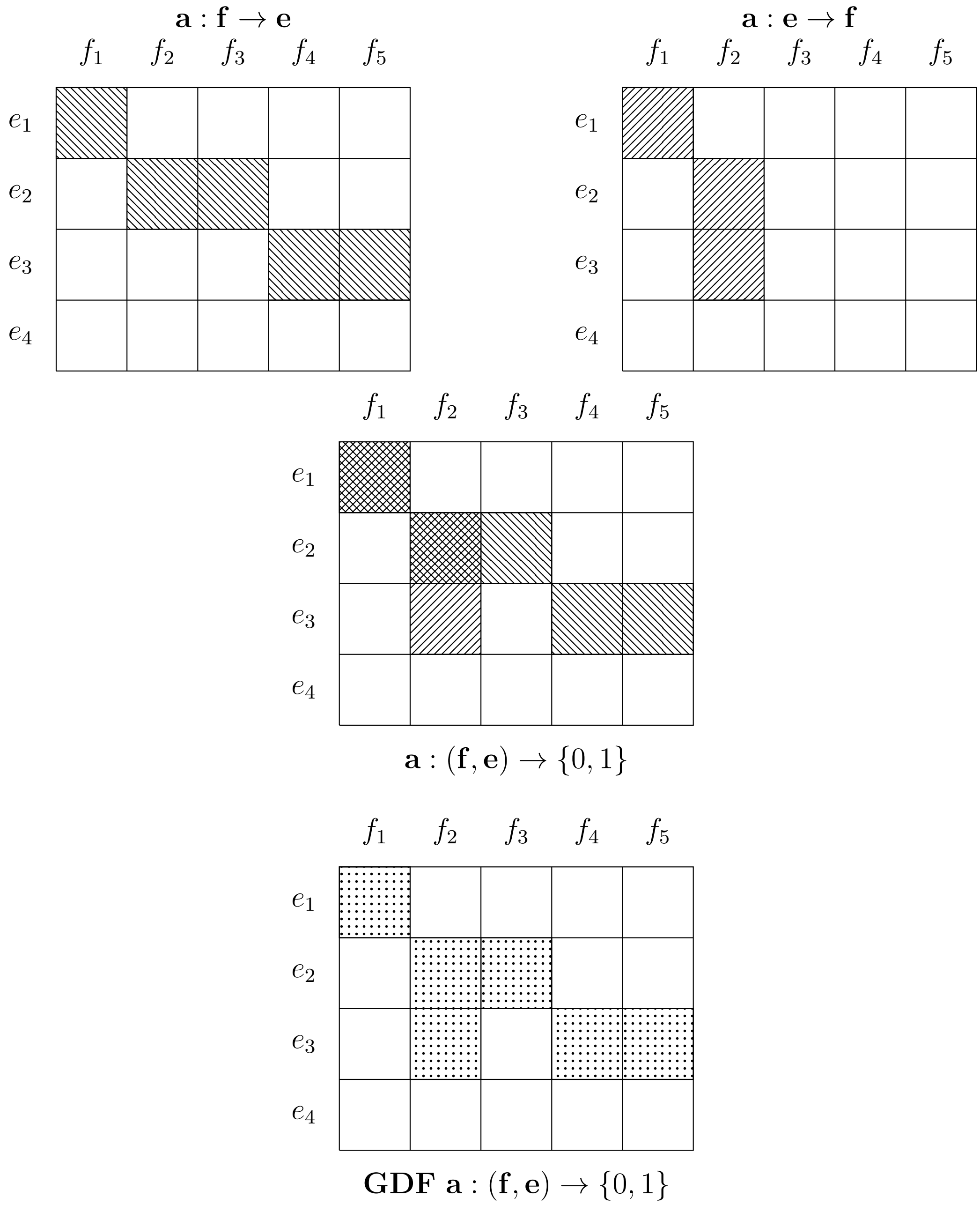


Figure 2.10: Example of union and intersection for symmetrization: The top left graph includes links 1-1, 2-2, 3-2, 4-3, 5-3 and the top right graph includes links 1-1, 2-2, 2-3. The middle graph displays union links 1-1, 2-2, 2-3, 3-2, 4-3, 5-3 and intersection links 1-1, 2-2. The bottom graph displays alignment links generated by GDF.

agreement with the expectation of a loss function which measures the disagreement between two models.

Ganchev et al. [2008a], Graça et al. [2010] propose a different approach, called Posterior Regularization (PR) [Ganchev et al., 2008b], that applies the constraints on the model posteriors by incorporating symmetry constraints. This is done by replacing the actual posterior distribution in the auxiliary function of the EM with a distribution that is (a) close to the posterior, (b) better matches the symmetry constraints. DeNero and Macherey [2011] propose to embed two-directional HMM aligners into a single model using dual decomposition instead of training two separate models. Sontag et al. [2010] share a similar idea of using dual decomposition as an approximate inference technique.

2.5 Summary

In this chapter, we presented the task of alignment for bitext at various levels from document-level to subword-level. This task aims to uncover hierarchically the hidden patterns between the text in the source language and its translation in another language. We highlighted the most outstanding models for document alignment, sentence alignment, and also sub-sentential alignment. In sub-sentential alignment, we discussed word alignment models under unsupervised learning and supervised learning; and the most interesting models for phrase alignment task. We also explored the constraints in structure alignment.

The focus of this dissertation is the word alignment, exposing the correspondences between the source words and target words in parallel sentences. Each language has a different way to express a concept, which is characterized by the compounding, agglutinative, and morphological aspects of its morphology. We described different types of mapping, which explained these differences between language pairs. Besides using word information, we showed that subword-based or character-based information is useful for word alignment. We described unsupervised generative word alignment models `IBM` [Brown et al., 1993b] and `HMM` [Vogel et al., 1996]. These models generate asymmetrical alignment which only consists of many-to-one links or null links. We explained the learning algorithm EM used in these models and also discussed Gibbs sampling used in Efmaral. We described the Viterbi algorithm and the Baum-Welch algorithm in the case of `HMM`. Different approaches to symmetrizing asymmetrical word alignments are presented: a built-in property approach and a heuristical post-processing approach.

Note that these generative models use unsupervised estimation techniques to build alignment links at the word level, relying on large collections of parallel sentences. Such approaches are typically challenged by low-frequency words, whose cooccurrences are poorly estimated and they also fail to take into account context information in alignment. Even though their performance (AER scores) seems fair for related languages (e.g. French-English), there is still much room for improving automatic alignments produced by standard tools such as `Giza++` [Och and Ney, 2003] or `Fastalign` [Dyer et al., 2013]. We also wonder if there are other hidden drawbacks of these models and how to uncover them. Therefore, we need a guide and also a collection of tools that help us to comprehensively observe all possible limitations of these traditional models. In the next chapter, a set of evaluation methods aims to focus on unaligned words, rare words, unknown words, function words, content words, and word orders, etc, will be proposed. We expect that these tools not only identify the limitations of these statistical models and also suggesting the appropriate approaches to improve them.

Chapter 3

Evaluating word alignments

For the task of word alignment, we recognize that there is no remarkable guide/tool that helps us to clarify all existing problems of each word alignment model. We believe that such guides/tools are necessary to evaluate new models and to understand what these new models improve. The implementation of these tools is available from `https://github.com/ngohoanhkhoa/Generative_Probabilistic_Alignment_Models`.

In this chapter, we first describe our training and test corpora (Section 3.1), reporting observations related to dataset size, sentence length, number of words, vocabulary, human reference alignment and also data pre-processing. We then explore a list of methods based on our bitext corpora to evaluate our models. Each method suggests the obstacles of each corpus that our models should overcome. We consider how to appropriately measure the performance of the models (Section 3.2). We present an analysis of common difficulties: unaligned words (Section 3.3), rare words (Section 3.6), unknown words (Section 3.7) and alignment types (Section 3.4), which is differently influenced by the morphology of each language. Word order (Section 3.5) and part-of-speech (Section 3.8) are also considered in our analysis. Our last question is about the symmetry that can be computed from asymmetrical alignments in both directions (Section 3.9). Note that we only show tables and figures that represent these obstacles while complete results are in [Ngo Ho, 2021, Appendix A].

Contents

Baselines We use these methods to evaluate two baselines implemented in statistical word alignment tools `Giza++` and `Fastalign`. All parameters of these models are set to their default values [Och and Ney, 2000, Dyer et al., 2013]. We train `IBM-1 Giza++` for 10 iterations (1^{10}), `HMM Giza++` (1^5H^{10}), `IBM-4 Giza++` ($1^5H^53^34^3$) [1] and `Fastalign` also for 10 iterations. Note that we concatenate the training and test data, which means that there is no unknown word for our baselines. We discuss in detail this issue in Section 3.7. Complete results of the baselines are in [Ngo Ho, 2021, Appendix A].

Notation If we use "English-Foreign", "En-XX" or "the direction" e.g., "the direction English-French", it means that the English language is on the source side and the French (Foreign) language is on the target side (representing the state side in the case of `HMM`). For asymmetric alignment models, they associate each word in a source side with exactly one word from the target side. Other cases such as "the language pair English-French" or only "English-French" mean that we mention both directions. Moreover for some confusing graphs/tables, we note which language is on the source or the target side in captions or in graph legends.

3.1 Parallel corpus

Our experiments consider six language pairs: English with French, German, Romanian, Czech, Japanese and Vietnamese. These languages belong to three language families, namely Indo-European languages (Czech, French, Romanian, German and English), Altaic language (Japanese) and Austroasiatic language (Vietnamese) [Lewis, 2009]. In detail, French and Romanian are in the family of Romance languages. German and English are classified into Germanic languages. Czech is one of Slavic languages. In our experiments, the writing system of Japanese uses logographs instead of the Latin alphabet that is used by all other languages. The Indo-European languages and Japanese are synthetic languages, which means that they use inflection or agglutination[2] to express syntactic relationships within a sentence. Vietnamese is an isolating language [Le et al., 2008] that has no inflectional morphology. This means that every word has exactly one form. Examples of these languages are displayed in Table 3.1.

Language	English sentence	Foreign sentence
German	but this is not what happens .	das stimmt nicht !
French	it is quite understandable .	ce est tout à fait compréhensible .
Romanian	what 's the story about ?	despre ce este vorba ?
Czech	i tried to examine myself .	pokusil jsem se sám se prohlédnout .
Vietnamese	it was a fine morning .	Đó là một buổi sáng đẹp trời .
Japanese	this is the biggest event in a year .	また この 法会 を 、 年間 最大 の 行事 と する 。

Table 3.1: Examples of English, French, German, Romanian, Czech, Vietnamese and Japanese parallel sentences

[1] x^y where x is a model name (1, H, 3, 4 represents model `IBM-1`, `HMM`, `IBM-3` and `IBM-4` respectively), y is a number of iterations.

[2] Inflection is the addition of morphemes to a root word that assigns grammatical property to that word, while agglutination is the combination of two or more morphemes into one word. The information added by morphemes can include indications of a word's grammatical category, such as whether a word is the subject or object in the sentence [Lewis, 2009, Dawson and Phelan, 2016].

3.1.1 Training corpus

Our Indo-European language training sets are mostly made of sentences from Europarl[3] [Koehn, 2005]: this is the case for French[4] and German. For Romanian, we use both the NAACL 2003 corpus[5] [Mihalcea and Pedersen, 2003] and the SETIMES corpus[6] used in WMT'16 MT evaluation. For Czech, the parallel data from News Commentary V11[7] [Tiedemann, 2012] is considered, while we use the preprocessed parallel data for Vietnamese in IWSLT'15 [Luong and Manning, 2015] and the Japanese data from The Kyoto Free Translation Task (KFTT[8]) [Neubig, 2011]. These corpora are tokenized with tools: the Moses toolkit[9] (for English, French, German and Czech), tokro[10] (for Romanian), KyTea[11] (for Japanese). Note that Vietnamese data is preprocessed using Vietnamese NLP toolkit[12]. In our experiments, we lowercase, clean and remove sentences with more than 50 words using the standard tools from the Moses toolkit.

Basic statistics for these corpora are in Table 3.2. English-French and English-German training data ($\geq$ 1.5M) are much larger than the rest (from 122K to under 400K). The French and German corpus are separated from the rest of the corpora and are a representative "large data" condition. Unsurprisingly, the vocabulary sizes of the German, Romanian and Czech corpora are substantially greater than the corresponding English, which contains a smaller number of inflected variants. The opposite pattern is found for our two other language families Japanese and Vietnamese, two synthetic languages with less inflectional variability than English. As an illustration of the difference between French and Vietnamese morphology, the verb "aller" has the different forms such as "vais", "vas", "va", "allons", "allez", "vont", ... whereas Vietnamese expresses the same concept using only one word "đi".

Training corpus	Number of sentence pairs	Number of words		Vocabulary		Char vocabulary	
		English	Foreign	English	Foreign	English	Foreign
English-French	~1.7M	~40M	~44M	106 322	112 734	111	115
English-German	~1.5M	~37M	~35M	96 898	311 582	218	235
English-Romanian	~250K	~5.6M	~5.8M	74 279	115 567	124	131
English-Czech	~182K	~4.2M	~3.8M	62 877	147 188	246	157
English-Japanese	~377K	~7.7M	~8.0M	156 107	126 246	2920	5766
English-Vietnamese	~122K	~2.1M	~2.5M	42 544	19 853	133	171

Table 3.2: Basic statistics for the training corpus after filtering based on the sentence length ($\leq$ 50 words)

3.1.2 Test corpus

For French and Romanian, we use data from the 2003 word alignment challenge[13] [Mihalcea and Pedersen, 2003]; the German test data is Europarl[14], while for Czech we use the corpus

[3] European Parliament Proceedings Parallel Corpus 1996-2011: https://www.statmt.org/europarl/

[4] To compare with related works, we also use the Hansards dataset (https://www.isi.edu/natural-language/download/hansard/index.html) with ~ 1.1M sentence pairs, which is smaller than the corpus from Europarl.

[5] https://web.eecs.umich.edu/ mihalcea/wpt/

[6] SETimes – A Parallel Corpus of English and South-East European Languages. http://nlp.ffzg.hr/resources/corpora/setimes/

[7] http://opus.nlpl.eu/News-Commentary.php

[8] http://www.phontron.com/kftt/

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

[10] https://perso.limsi.fr/aufrant/software/tokro

[11] http://www.phontron.com/kytea/

[12] https://vlsp.org.vn/wiki/tools; https://github.com/manhtai/vietseg

[13] https://web.eecs.umich.edu/ mihalcea/wpt/

[14] http://www-i6.informatik.rwth-aachen.de/goldAlignment/

described in [Mareček, 2016][15]. The Japanese test data is also from KFTT[16]. The test corpus for Vietnamese is generated from the EVBCorpus[17]. We also use the 2015 word alignment challenge[18] [Mihalcea and Pedersen, 2003] for Romanian (English-Romanian Dev) to select appropriate configurations for our models.

Each test corpus includes a parallel data and an alignment set which shows word correspondences for each sentence pair. For a sentence pair (made of a source sentence with J words and a target sentence with I words), an alignment link takes the form $j - i$, where j, i are respectively the index of source and target word. We set the index of the first word to 1 in each sentence. For example, Figure 3.1 displays the links 1-1, 2-2, 3-3, 3-4, 3-5, 4-6, 5-7 between five source words and seven target words.

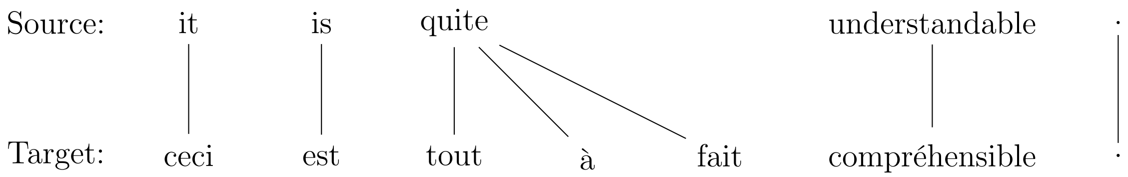


Figure 3.1: Example of an alignment set containing links 1-1, 2-2, 3-3, 3-4, 3-5, 4-6, 5-7 between five source words and seven target words.

Basic statistics for these corpora are in Table 3.3. We also report the number of words never seen in the training data (unknown words) and the corresponding number of unknown types in parentheses. The test datasets for Czech, Japanese and Vietnamese are considerably larger than the other test corpora. Recall that for these languages we have a comparatively small amount of train data (see Table 3.2). This explains the large number of unknown words in the case of Czech and Vietnamese.

Test corpus	Number of sentence pairs	Number of words		Number of unknown words	
		English	Foreign	English	Foreign
English-French	447	7 020	7 761	157 (60)	64 (50)
English-German	509	10 413	9 945	15 (15)	58 (58)
English-Romanian	246	5 455	5 315	36 (30)	62 (55)
English-Czech	2 501	59 724	52 881	1 599 (843)	2 546 (1 769)
English-Japanese	1 235	30 822	34 403	560 (418)	240 (190)
English-Vietnamese	3 447	70 049	94 753	4 855 (1 977)	2 818 (903)
English-Romanian Dev	200	4 562	4 365	1 (1)	15 (15)

Table 3.3: Basic statistics for the test corpora

3.1.3 Alignment links

We report the number of alignment links in the test corpora in Table 3.4. These links are the human reference alignments including sure and possible alignments. An example of these links is in Figure 3.2. The number of word pairs is the total number of alignment links possibly generated, i.e., for each sentence, this number is equal to $I * J$ where I and J are respectively the length of source and target sentence. An observation is that Romanian, Japanese and Vietnamese[19] corpora only contain sure links. To clarify our analysis, a non-alignment link is

[15] https://lindat.mff.cuni.cz/repository/xmlui/handle/11234/1-1804
[16] http://www.phontron.com/kftt/# alignments
[17] https://code.google.com/archive/p/evbcorpus/
[18] https://web.eecs.umich.edu/ mihalcea/wpt05/
[19] The human reference for English-Vietnamese does not contain links between punctuation.

a link $i - j$ which does not exist in the alignment set, but the source i or target word j could be aligned to another target/source word.

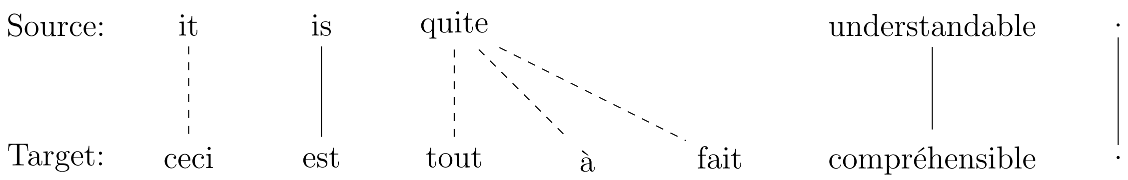


Figure 3.2: Examples of sure (2-2, 4-6, 5-7) and fuzzy (1-1, 3-3, 3-4, 3-5) alignment links.

Test corpus	Number of word pairs	Number of alignment	
		Sure	Possible
English-French	143 000	4 038	13 400
English-German	240 263	9 612	921
English-Romanian	160 509	5 991	0
English-Czech	1 660 327	44 292	23 131
English-Japanese	1 398 756	33 377	0
English-Vietnamese	2 507 568	81 748	0
English-Romanian Dev	132 258	5 035	0

Table 3.4: Basic statistics for the links in the test datasets

3.2 How to score predicted alignments ?

We use the AER [Och, 2003], accuracy (ACC), F-score (F1), precision (PRE) and recall (REC) [Jardine and van Rijsbergen, 1971, Derczynski, 2016] as measures of performance. AER is based on a comparison between predicted alignment links (A) and a human reference including sure (S) and fuzzy links. The set P contains in addition to these sure links, these fuzzy links ($S \subseteq P$). This score is defined as an average of the recall and precision taking into account the sets P and S. These scores are defined as:

$$\text{AER} = 1 - \frac{|A \cap S| + |A \cap P|}{|A| + |S|} \tag{3.1}$$

$$\text{Precision} = \frac{|A \cap P|}{|A|} \tag{3.2}$$

$$\text{Recall} = \frac{|A \cap P|}{|P|} \tag{3.3}$$

$$\text{F-score} = \frac{2 * |A \cap P|}{|A| + |P|} \tag{3.4}$$

$$\text{Accuracy} = \frac{|A \cap P| + |(U - A) \cap (U - P)|}{|U|} \tag{3.5}$$

where A is the set of predicted alignments, U is the set of all alignments possibly generated. Because of $S \subset P$, the unbalance between precision and recall is not penalized by AER but the F-score is [Fraser and Marcu, 2007]. In other words, the AER score is easy to game because adding more fuzzy links can only worsen the AER and predicting fewer links is the right thing to do here (favoring precision over recall), leading to a bias in alignment evaluation. Note that the Romanian, Japanese and Vietnamese reference data only contain sure links ($S = P$); in this case, AER and F-measure are deterministically related.

In our analysis, we also observe the confusion matrix [Derczynski, 2016] where P and N respectively represent the alignment link and the non-alignment link. True Positive (TP) is the number of correct alignment links, False Positive (FP) is the number of incorrect alignment links, True Negative (TN) is the number of correct non-alignment links and False Negative (FN) is the number of incorrect non-alignment links.

Two ways of training the baselines It is possible to merge test and training corpus, which implies there is no unknown word. We use this way of training in all of our analyses. A more realistic case is to separate test and training corpus where we introduce a UNK token in test corpus. We observe model performance for these two cases.

The scores of our baselines are in [Ngo Ho, 2021, Appendix A.1]. In the case of concatenating test and training corpus, there are two main observations that pose a challenge about a model balancing the precision and the recall.

- A drawback of AER: `IBM-4 Giza++` tends to favor precision over recall, which yields a better AER than `Fastalign` and `HMM Giza++` but a worse F-score. This can be appropriate for English-French that includes a large number of possible links (Table 3.5). This situation is not found in other language pairs.

- `Fastalign` outperforms `IBM-4 Giza++` in the case of Czech-English (Table 3.6), English-Japanese and English-Vietnamese in both directions. This is explained by the reduction of the number of incorrect non-alignment links (FN), e.g. -3805 (Czech-English) links as can be seen in Figure 3.4 on page 51.

Compared with the previous case, the first observation is that separating test and training corpus worsens the performance of `Fastalign` and `IBM-4 Giza++`. The loss can be large e.g. about +7 AER in the direction English-Czech. However, for `IBM-1 Giza++`, we see an improvement in the case of the language pair English-French, the direction German-English, Czech-English, English-Romanian and English-Vietnamese. This improvement can be found in `HMM Giza++` for the language pair English-French, English-German and English-Romanian and the direction English-Czech.

Models	English-Foreign					Foreign-English				
	AER	F1	PRE	REC	ACC	AER	F1	PRE	REC	ACC
	Concatenation									
`IBM-1 Giza++`	40.1	26.7	71.55	16.41	89.01	33.9	36.49	59.24	26.37	88.81
`Fastalign`	15.19	44.98	82.5	30.92	90.78	16.23	46.32	80.08	32.58	90.79
`HMM Giza++`	11.99	45.18	86.12	30.62	90.94	11.97	45.98	85.2	31.49	90.98
`IBM-4 Giza++`	10	44.43	90.61	29.43	91.02	9.64	45.43	89.58	30.43	91.08
	Replacing unknown words with the token UNK									
`IBM-1 Giza++`	30.97	36.89	64.26	25.87	89.21	33.32	36.99	60.06	26.73	88.9
`Fastalign`	15.33	44.91	82.41	30.86	90.77	16.41	46.21	79.93	32.5	90.77
`HMM Giza++`	10.83	45.82	87.69	31.01	91.06	11	46.66	86.53	31.94	91.09
`IBM-4 Giza++`	15.02	41.41	88.94	26.99	90.69	12.44	43.4	88.81	28.71	90.87

Table 3.5: Alignment error rate (AER), accuracy (ACC), F-score (F1), precision (PRE) and recall (REC) for English-French

Extrinsic measures Besides these methods that directly evaluate alignment quality, we can evaluate the alignment performance through the results of downstream tasks using word alignment. Several important tasks are phrase-based translation (Section 2.2.3.2), machine

Models	English-Foreign					Foreign-English				
	AER	F1	PRE	REC	ACC	AER	F1	PRE	REC	ACC
	Concatenation									
`IBM-1 Giza++`	45.09	46.75	50.4	43.59	95.97	48.47	42.88	49.17	38.02	95.89
`Fastalign`	25.75	64.09	70.98	58.42	97.34	25.3	62.86	73.13	55.13	97.36
`HMM Giza++`	27.86	61.22	70.81	53.92	97.23	30.38	57.28	69.26	48.83	97.04
`IBM-4 Giza++`	20.92	65.7	79.48	56	97.63	26.5	59.81	75.58	49.48	97.3
	Replacing unknown words with the token UNK									
`IBM-1 Giza++`	45.51	46.42	50.05	43.28	95.94	46.08	44.87	51.45	39.79	96.03
`Fastalign`	26.56	63.48	70.2	57.93	97.29	26.18	62.14	72.29	54.48	97.3
`HMM Giza++`	27.86	61.23	70.96	53.86	97.23	32.21	56.02	67.32	47.97	96.94
`IBM-4 Giza++`	28.56	58.94	72.79	49.51	97.2	32.48	54.86	69.59	45.28	96.97

Table 3.6: Alignment error rate (AER), accuracy (ACC), F-score (F1), precision (PRE) and recall (REC) for English-Czech

translation with/without attention mechanisms Mi et al. [2016], Liu et al. [2016], Chen et al. [2016], Alkhouli and Ney [2017], bilingual dictionary extraction [Liu et al., 2013, Héja, 2010], noise filtering from translation memories, parallel corpora cleaning ([Pham et al., 2018]), automatic quality estimation [Wisniewski et al., 2013, Stymne et al., 2014, Specia et al., 2018], etc. For machine translation, there are several scores which can reflect model performance for the word alignment task such as BLEU (Bilingual Evaluation Understudy) [Papineni et al., 2002], METEOR (Metric for Evaluation of Translation with Explicit Ordering) [Banerjee and Lavie, 2005], WER (Word Error Rate) [Klakow and Peters, 2002], ROUGE (Recall-Oriented Understudy for Gisting Evaluation) [Lin, 2004], NIST (National Institute of Standards and Technology) [Doddington, 2002], etc.

3.3 Issues with unaligned word

For some language pairs, it is difficult to know a word that should be unaligned or aligned. This creates a disagreement between annotators. For example, English pronouns can be kept unaligned or align with the Czech verbs. A similar situation arises with Czech reflexive pronouns that have no real equivalents in English [Čmejrek et al., 2004, Kruijff-Korbayová et al., 2006]. In addition, for machine translation systems, Zhang et al. [2009] show that the presence of unaligned words causes extraction of noisy phrases, leading to insertion and deletion errors in the translation output. For the generative IBM models, they process words that likely have no translation by introducing a NULL word on the generating side. All words on the source side without a proper target translation would then be generated by that NULL word [Schulz et al., 2016]. It is clear that the role of unaligned words is important. Therefore, we need a detailed analysis for this type of words in word alignment.

Statistics for the number of unaligned words are in Table 3.7. We compute also the average ratio of the number of unaligned words to the total number of words for one sentence, which makes Japanese ($\sim$23% and $\sim$18%) and Vietnamese ($\sim$32% and $\sim$16%) different from the rest ($\leq$13%). In fact, at least a quarter of English words do not align with any Japanese/Vietnamese word. The ratios of above 10% witnessed in German and Romanian, also underline the unaligned word issue for these languages. An example of unaligned words for English-Vietnamese is in Figure 3.3.

We collect correct/incorrect alignment/non-alignment links and unaligned words on both sides to observe the alignment errors for each baseline. Complete results are in [Ngo Ho, 2021, Appendix A.2] for alignment links and [Ngo Ho, 2021, Appendix A.3] for unaligned words. Details for the English-Czech language pair are in Figure 3.4.

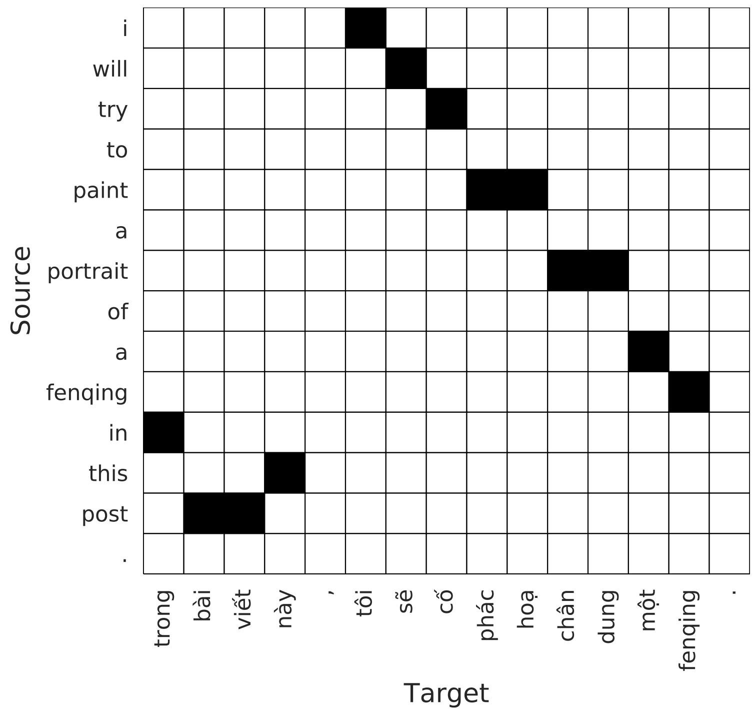


Figure 3.3: Example for unaligned English words ("to", "a", "of" and ".") and Vietnamese words ("," and "."). The ratio of unaligned English and Vietnamese word is $\frac{4}{14}$ and $\frac{1}{15}$ respectively.

Test corpus	Number of unaligned words		Ratio of unaligned word %	
	English	Foreign	English	Foreign
English-French	327	349	4.21	4.61
English-German	858	1 272	8.06	13.0
English-Romanian	507	491	11.9	10.5
English-Czech	3 326	4 070	6.18	6.84
English-Japanese	7 965	6 352	23.6	18.1
English-Vietnamese	22 367	15 785	32.0	16.6
English-Romanian Dev	528	471	12.4	10.0

Table 3.7: Basic statistics of unaligned words for the test corpora

Regarding `IBM-1 Giza++` [Moore, 2004], we observe that too few source words are linked to the NULL token on the target side, e.g., the number of unaligned words is significantly smaller than the reference as can be seen in English-Czech (Figure 3.5). This can be explained by the structure of `IBM-1` including only one NULL token on the target side. The opposite trend is observed in the case of English vs French, Romanian and Vietnamese. Most of their unaligned English words are function words and clearly incorrect. The problem of function words is discussed in Section 3.8.

Our most complex baselines `IBM-4 Giza++` does not generate more correct links than other models, but simply removes the incorrect links. This situation yields a small number of correct non-alignment links but also creates a large number of incorrectly unaligned words in Figure 3.5. We also recognize that the distortion model is more complex, there are more incorrect unaligned source words. The figure again highlights the unbalance between precision and recall of our baselines, which requires a better approach for unaligned words. Similar patterns can be observed in the other corpora.

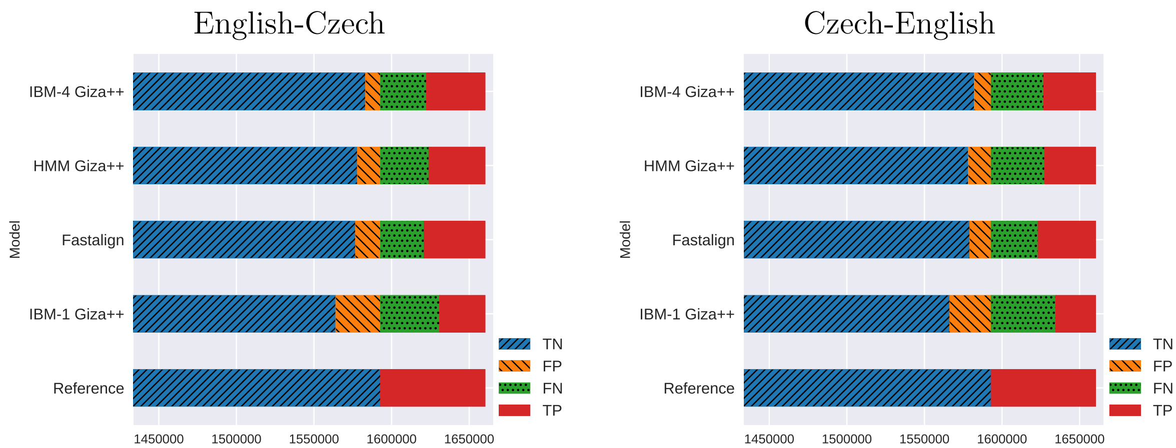


Figure 3.4: Results of our baselines: Alignment links for the direction English-Czech and the direction Czech-English

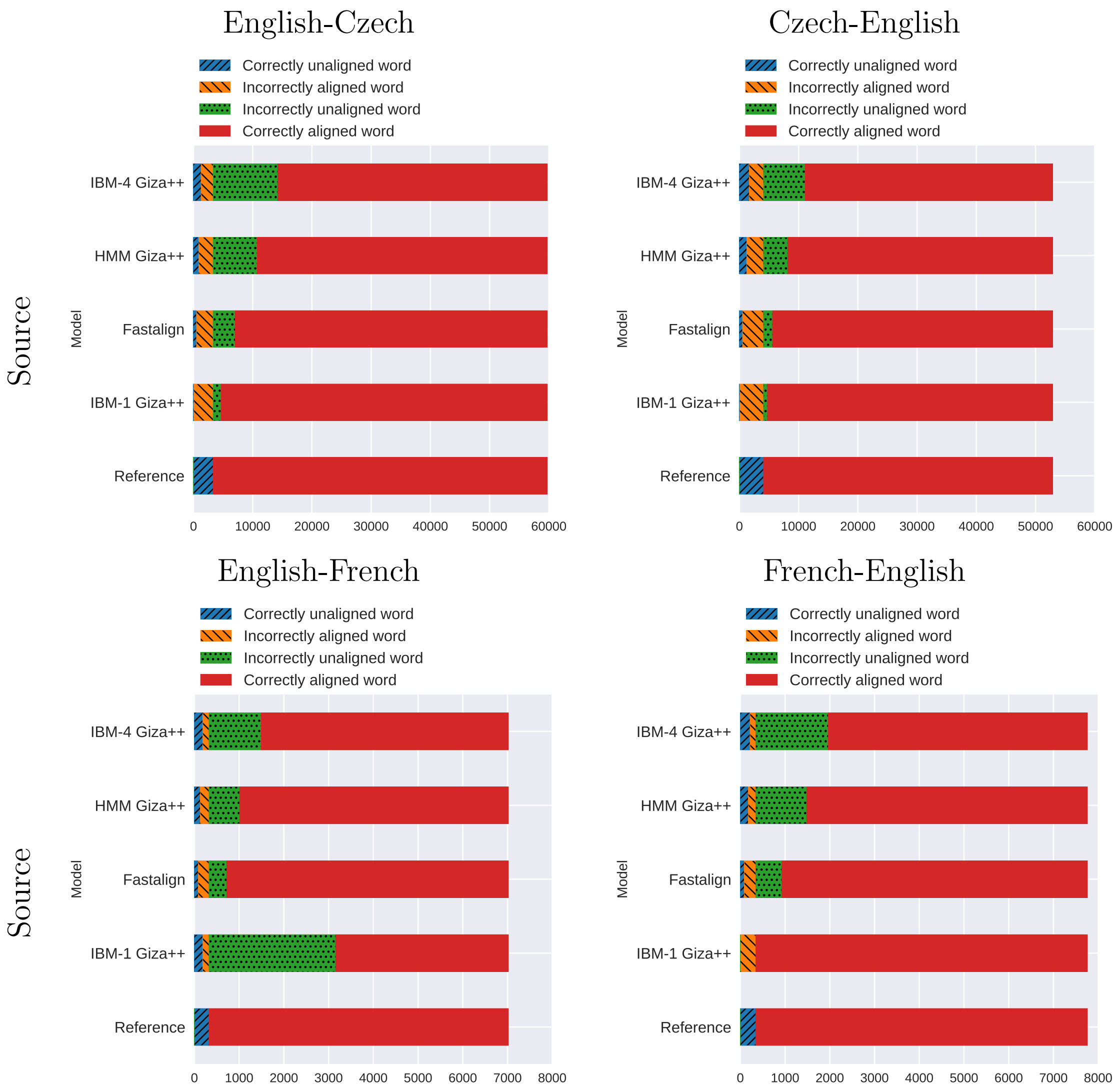


Figure 3.5: Results of our baselines: Unaligned words for the direction English-Czech/Czech-English and the direction English-French/French-English

3.4 Weaknesses of asymmetrical alignments

Alignment links are categorized by their types as one-to-one, one-to-many, many-to-one and many-to-many (Section 2.3.1). Some models are impossible to directly predict all alignment link types. For example, the above-mentioned generative alignment models can generate neither one-to-many nor many-to-many links. It should be noticed that the distribution of these types in the human reference alignments can describe the requirements of each language pairs for our models. Therefore, we discuss how to count the number of these alignment link types and explain how to faithfully report the performance for these link types.

In the case of one-to-one links, there is only one source word aligning to only one target word. For one-to-many/many-to-one, there are at least two target/source words aligning to only one word in the source/target side respectively. These two types are characterized by two numbers, the left number represents the number of source words and the right number indicates the number of target words. The number of one-to-many/many-to-one links is also the number of target/source words. The case of many-to-many is a complex issue, clarified in Figure 3.6. Many-to-many contains an extra value, the number of many-to-many links in parentheses. An example is in Figure 3.6. Another number (%) is the ratio of the number of links for an alignment type to the total number of links.

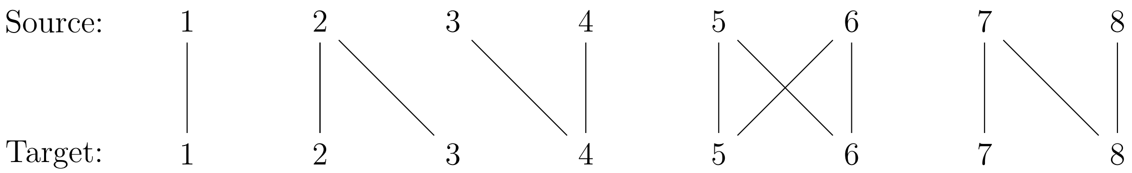


Figure 3.6: Example of type alignment: link 1-1 is one-to-one. links 2-2, 2-3, 7-7 are one-to-many. link 3-4, 4-4, 8-8 are many-to-one. four links 5-5, 5-6, 6-5, 6-6 are many-to-many. link 7-8 could be both one-to-many and many-to-one link, it is counted as a many-to-many link

Basic statistics of these alignment types are in Table 3.8. For example, English-French has 3 174 one-to-one links, 1 120 one-to-many links involving 549 English words and 1 120 French words. We observe that the English-French corpus contains a large number of many-to-many links (∼12.6K links) compared to the other types of alignment. This suggests that models that can generate many-to-many links significantly benefit from this type of alignments. This is also the case of one-to-many links for English-Vietnamese/Japanese (Figures 3.7), many-to-one links for English-Czech. For Vietnamese, the difference between one-to-many and the other alignment types is very large. It is because an English word is often translated into more than two Vietnamese words[20] [Le et al., 2008]. Therefore, subword-based models for English seem to be useful when an English source word aligns with several Vietnamese/Japanese words (see an example in Figure 3.7). We discuss the technique of using subwords in Chapter 6.

Test corpus	one-to-one	one-to-many	many-to-one	many-to-many
English-French	3 174 (18.2%)	549 - 1 120 (6.4%)	478 - 232 (2.7%)	2 492 - 2 886 (12 666) (72.6%)
English-German	6 024 (57.2%)	635 - 1 333 (12.6%)	2 769 - 1 209 (26.3%)	127 - 107 (407) (3.8%)
English-Romanian	2 933 (48.9%)	481 - 1010 (16.8%)	1 224 - 569 (20.4%)	310 - 312 (821) (13.7%)
English-Czech	27 703 (41.1%)	4 325 - 7 734 (11.5%)	20 609 - 10 501 (30.6%)	3 761 - 2 873 (11 377) (16.9%)
English-Japanese	12 687 (38.0%)	4 323 - 11 711 (35.1%)	4 252 - 1 908 (12.7%)	1 595 - 1 745 (4 727) (14.2%)
English-Vietnamese	21 455 (26.2%)	23 806 - 55 315 (67.6%)	635 - 294 (0.77%)	1 786 - 1 904 (4 330) (5.3%)
English-Romanian Dev	2 407	345 - 758	945 - 426	337 - 303 (924)

Table 3.8: Basic statistics of alignment type for the test corpora.

Complete statistics of alignment types generated by our baselines are in [Ngo Ho, 2021, Appendix A.4]. The distortion model helps to generate more one-to-one links as can be seen

[20] In Vietnamese lexicon, these Vietnamese words can be combined to a token, called compound word consisting of more than two syllables.

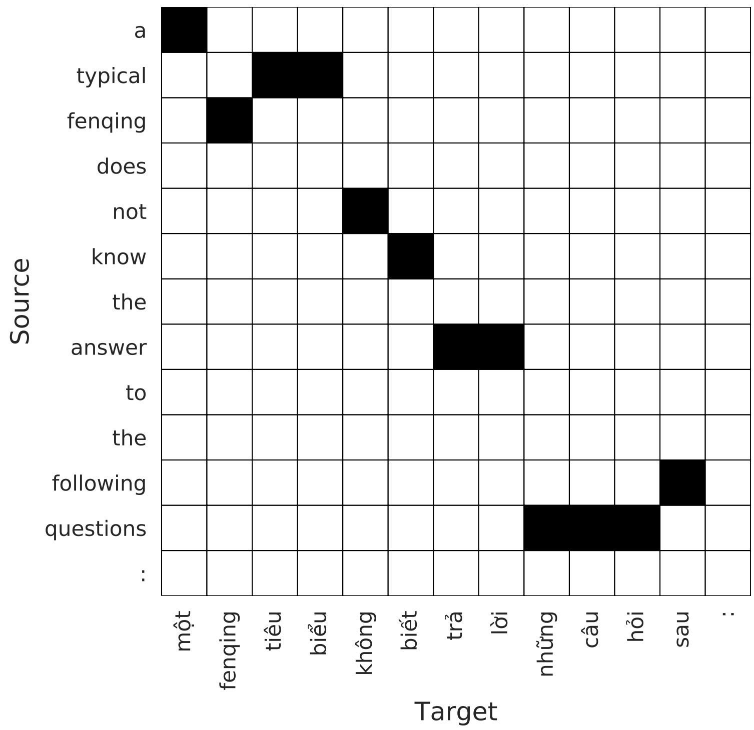


Figure 3.7: Example of one-to-many alignment links for English-Vietnamese: “typical"-[“tiêu", “biểu"], “answer"-[“trả", “lời"] and “questions"-[“những", “câu", “hỏi"].

for `Fastalign`, `HMM` and `IBM-4` (e.g. Figure 3.8). We see in Figure 3.9 that most of these links are correct. Moreover, the reduction of the number of alignment links mostly concerns the many-to-one type, which is harmful in the case of corpora containing a large number of this type (e.g., English-Czech with more than 20K links). This tendency is also observed for the other corpora.

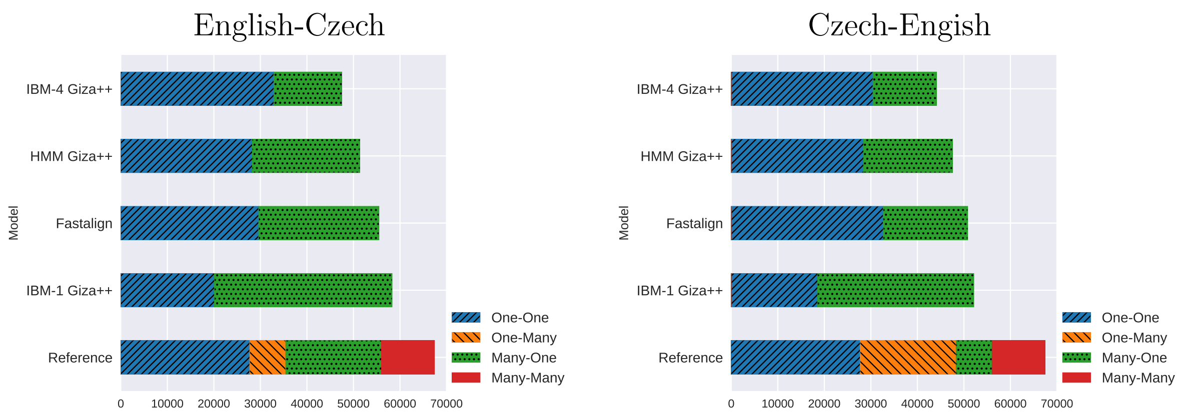


Figure 3.8: Results of our baselines: Alignment types for English-Czech

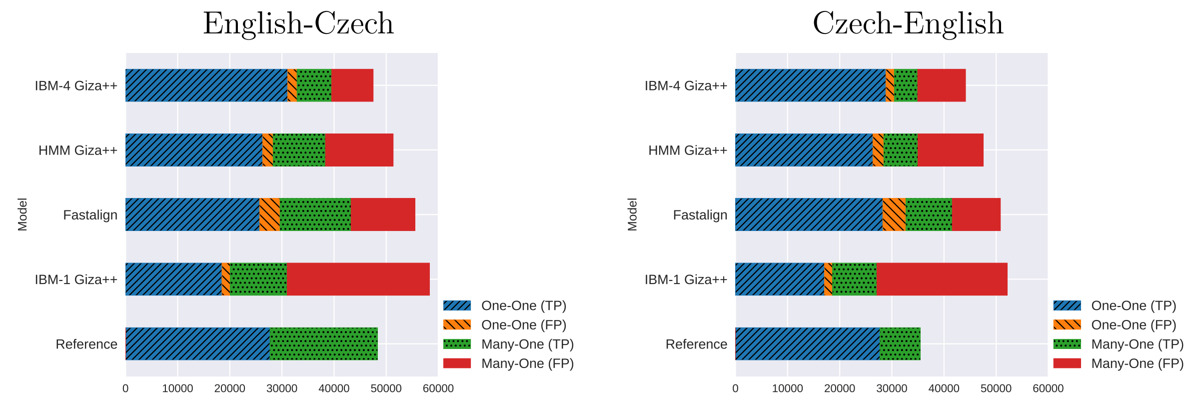


Figure 3.9: Results of our baselines: Alignment types for English-Czech

3.5 Monotonicity and Distortion

One of the properties of asymmetrical models is that each source word can be linked exactly once and linking to NULL token refers to not assigning any target word. This requires a model that captures word order divergences: rearranging all target words based on a source word order and determining unaligned source words. Our proposed models are mainly based on HMM model which includes first-order dependencies between adjacent links. Therefore, we explore general patterns of word order divergence by observing jumps of words in a sentence in relationship to its translation. We count the number of jumps as a function of jump width.

For the languages using the same typological system as English, we expect that models select target positions that are close to the diagonal of the alignment matrix (i.e., forward jumps). Moreover, we also expect crossing links (i.e., backward jumps) when there are differences between two typological systems. For example, English clauses mostly follow a SVO (subject-verb-object) word order while SOV (subject-object-verb) is the canonical word order in Japanese.

Determining the ”reference” jump is a complex issue, as the reference may contain cases of one-to-many, many-to-one alignments and many-to-many, yielding a set of possible reference jump values. In our analysis, we use the median, the minimum and the maximum of all possible word locations to compute jump values. An example of alignment and the associated jumps is in Figure 3.10.

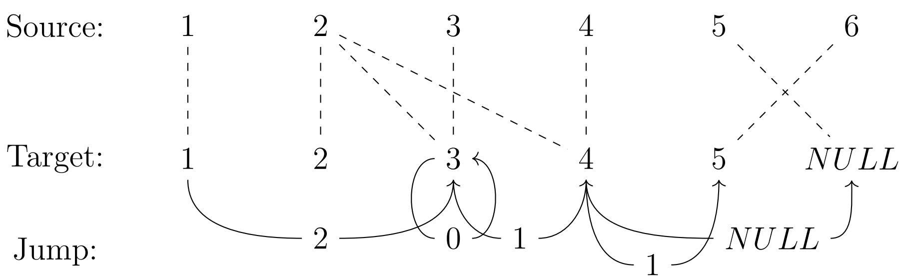


Figure 3.10: Example of the jumps in a target sentence: We see that the second source word is linked to the 2nd, 3rd and 4th target words. The median, the minimum and the maximum value is respectively 3,2 and 4. In the case of using median values, there are jumps of width 2, 0 and 1 and a jump to a NULL token.

For the Indo-European languages, most jumps are forward jumps, which highlights that these languages share similar word orders. We also recognize the prevalence of the short jumps (0 or 1) which corresponds to two main patterns:

- Most of the links have a jump of length 1, which is found in English-German (Figure 3.12), English-Romanian and English-Czech on both sides. This trend underlines the monotonicity in the alignment of these languages, suggesting a large number of near diagonal alignment links.

- Jump of 0 and 1 obtain similar numbers of links. This trend is only found in the case of English-French on both sides (e.g. Figure 3.12), which may be due to a large number of many-to-many links (Table 3.8). An example of such alignment links is in Figure 3.11.

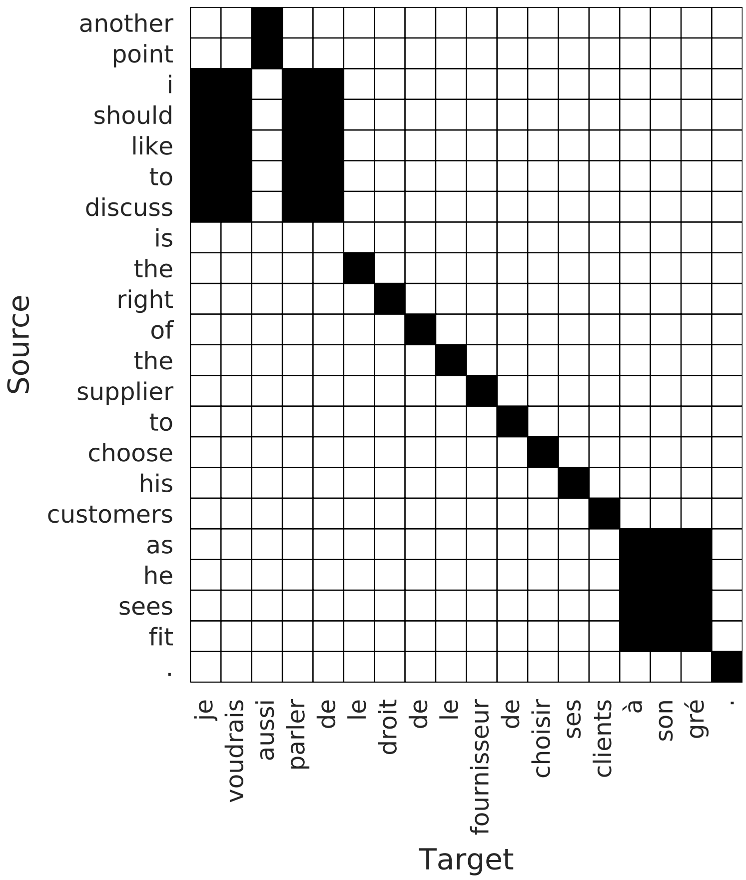


Figure 3.11: Example of alignment links for English-French: the word groups ["i", "should", "like", "to", "discuss"] and ["je", "voudrais", "parler", "de"]; ["as", "he", "sees", "fit"] and ["à", "son", "gré"]

In the case of Asian languages, we notice the opposite pattern e.g. Figure 3.12. The number of links with a jump value of 0 is larger than the correspondence of 1. This is explained by the frequency of one-to-many links in the alignment set (Table 3.8). We also observe a large number of Vietnamese and Japanese words jumping to NULL tokens, highlighting again the high ratio of unaligned English words (Table 3.7). An example of alignment links for English-Vietnamese is in Figure 3.13. Moreover, we recognize the crossing links with a large number of backward jumps in the case of Japanese, due to different word orders between English and Japanese (SVO and SOV).

To evaluate the behavior of our baselines, we set the reference jump as the median of all possible jumps. We first collect the number of jumps [Ngo Ho, 2021, Appendix A.5.1] and then correct/incorrect jumps [Ngo Ho, 2021, Appendix A.5.2]. In this case, we consider as correct a jump that creates a correct link. To analyze the distortion errors, we plot the confusions of the distortion models. In these representations, each cell (k, k') counts the number of times the model predicted a jump of k position, whereas the reference jump for that position was k'. We only count an error for each missing or erroneous jump value if the previous target word location is correctly predicted. These matrices are represented as heat-maps (see some examples in Figure 3.15): The darker cell, the greater the number of confusions.

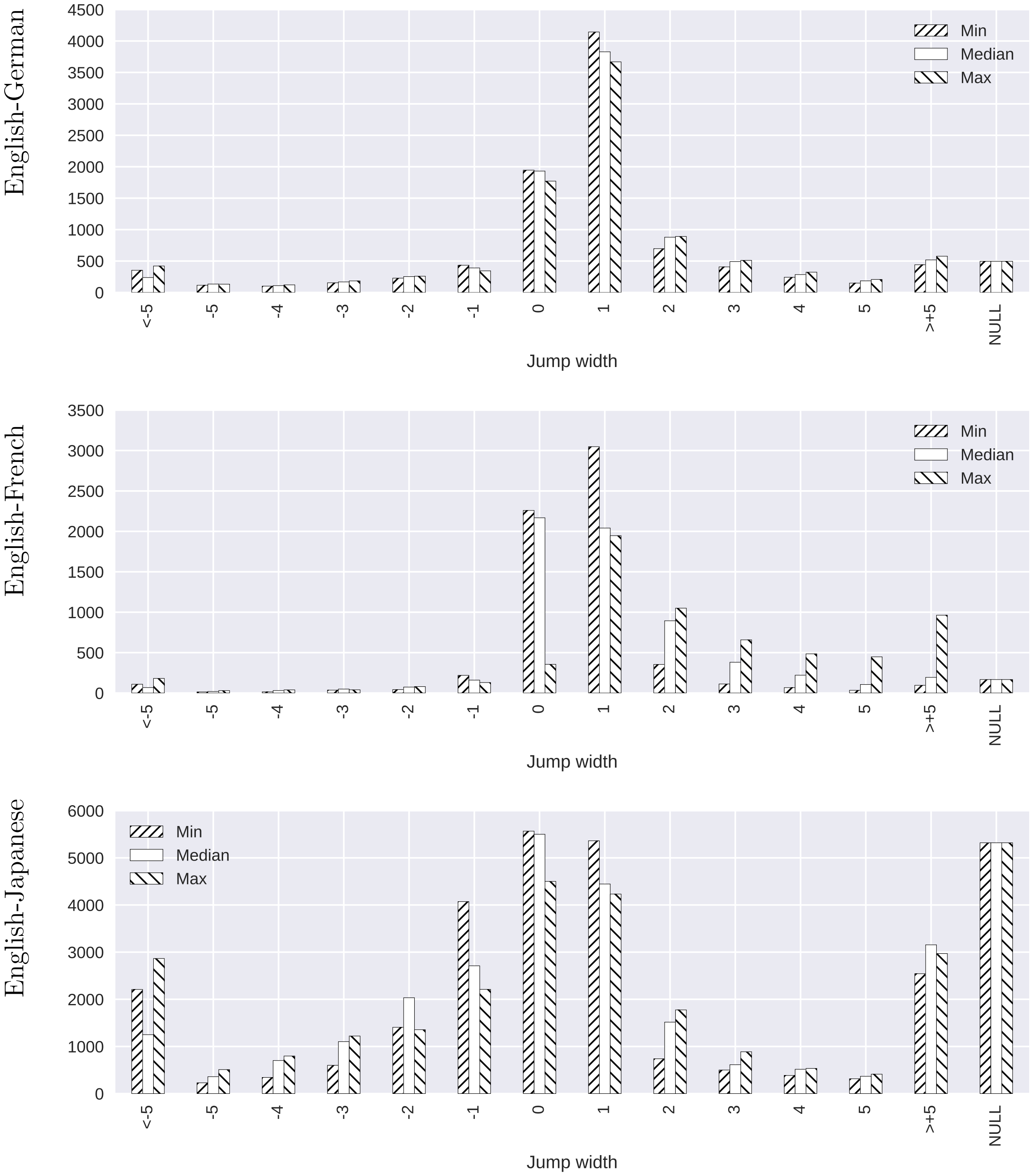


Figure 3.12: Jump patterns for the directions English-German, English-French and English-Japanese reference word alignments. The x axis shows the jump width and the y axis shows the number of alignment links.

A similar trend of English words in the reference is reproduced by our baselines [Ngo Ho, 2021, Appendix A.5.1]: There is a prevalence of short jumps of length 1 for our four Indo-European languages and short jumps of length width 0 in our two Asian languages.

- IBM-1: We notice that `IBM-1 Giza++` generates a large number of long jumps (Jump < -5 and > 5) for English words in all corpora. Half of these jumps are incorrect because the correct jump value should be 0, 1 or jump to the NULL token (Figure 3.14). This is also true for German and Czech. Besides the short jumps, for French and Romanian words, `IBM-1` also creates a large number of jumps to NULL tokens, only a small portion of which is correct (Figure 3.14). We also notice that `IBM-1 Giza++` creates a substantial number of incorrect jumps of value 0 which is even larger than the reference number in the case of Japanese and Vietnamese words (see Figure 3.14).

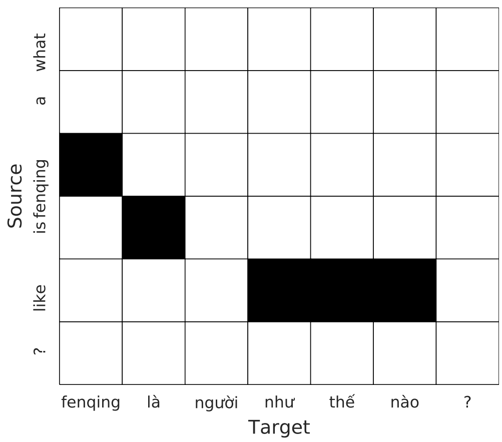


Figure 3.13: Example of alignment links for English-Vietnamese: the word "like" is linked to the Vietnamese words "như", "thế" and "nào"; the words "a", "what" are unaligned words.

- More complex baselines with a distortion model: Most of the incorrect links belong to the jump 0 and the jump to NULL token. This situation is even worse in the case of `Giza++` models as can be seen in Figure 3.15

In general, our baselines tend to over-predict a few of jump widths, failing to detect complex distortion patterns. This was a known problem for distortion models in SMT.

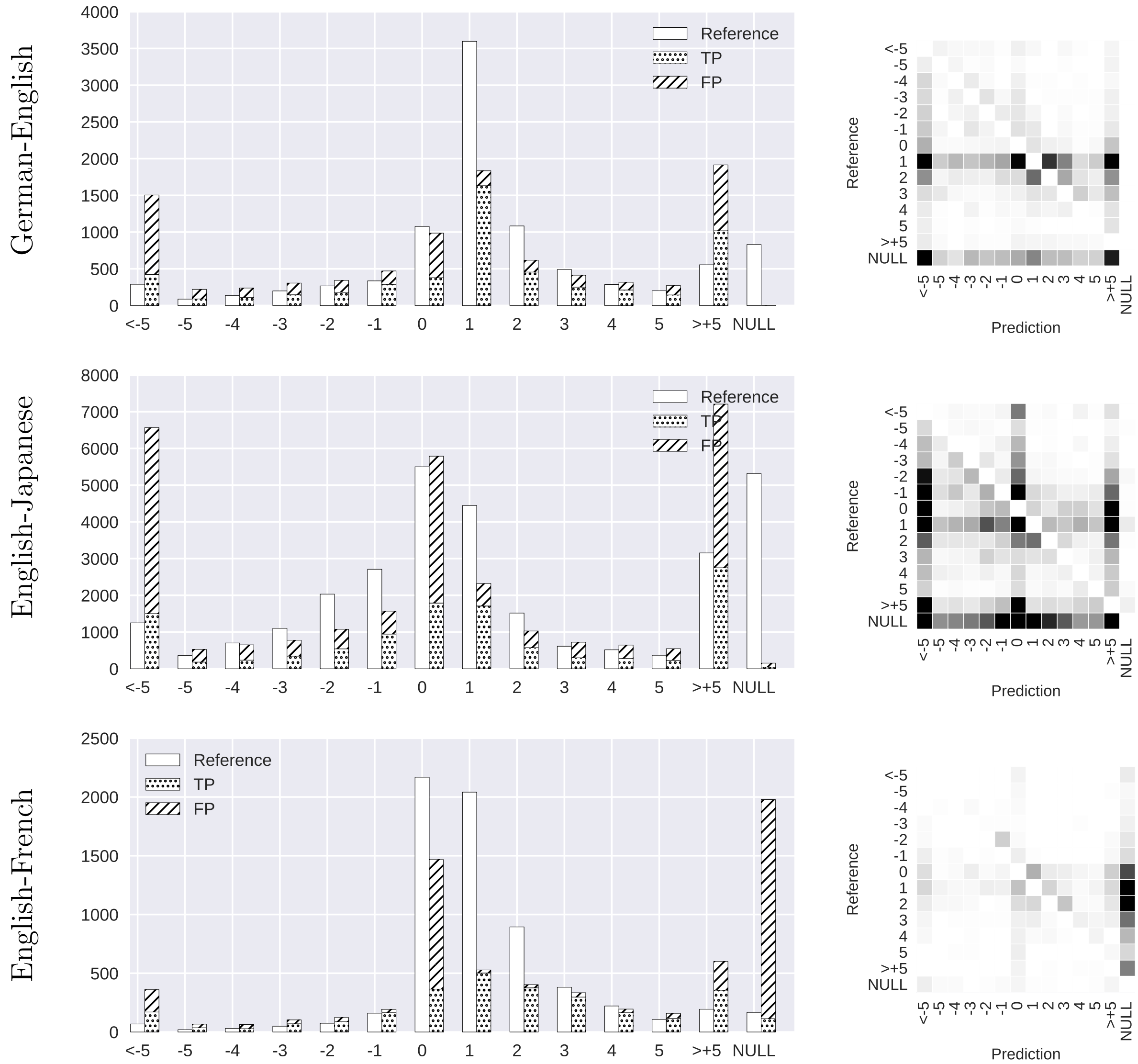


Figure 3.14: `IBM-1 Giza++`: Correct (TP) and incorrect (FP) jumps for English words (the direction German-English), Japanese words (the direction English-Japanese) and French words (the direction English-French) on the left graph. Confusion matrices on the right graph: The darker the cell, the greater the number of confusions.

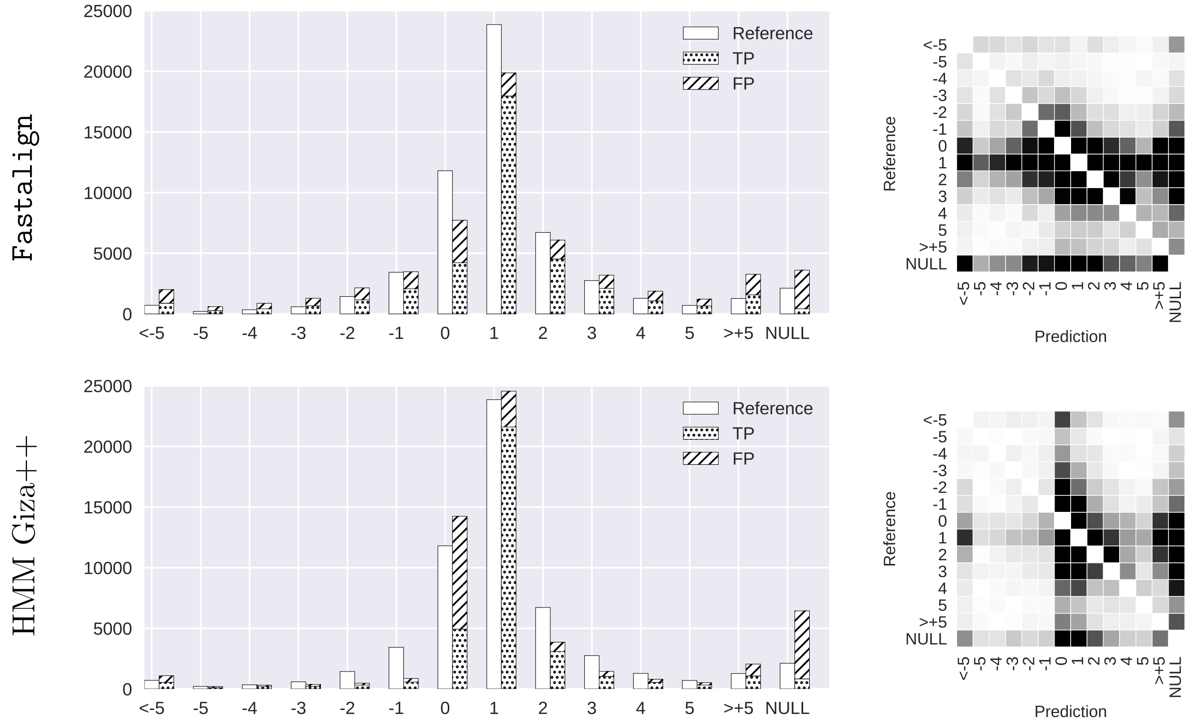


Figure 3.15: `Fastalign` and `HMM Giza++` for English-Czech: Correct (TP) and incorrect (FP) jumps for Czech words on the left graph. Confusion matrices on the right graph: The darker the cell, the greater the number of confusions.

3.6 Is there a problem with rare words?

One well-known issue with `Giza++` and `Fastalign` is the so-called ”garbage collector problem” (GCP) causing rare words in the target language to be misaligned to many source words [Brown et al., 1993a, Moore, 2004]. The definition of this problem is slightly different in [Wang et al., 2015b]: the authors present as a tendency of rare words to align with untranslated words. This is due to the maximization of the likelihood during EM: rare words often have a lot of spare mass in their conditional distribution and it is beneficial to align many source words to a rare target word. An example for a Romanian rare word "sireturi" is in Figure 3.16. This word is erroneously linked to the English words "must", "demoiselle", "generate", "such", "low", "-" and "down". As a general rule, rare source words should with high probability align with rare targets e.g., which signals "hobnobbing", a rare English word, as the right alignment for “sireturi” [Lardilleux et al., 2011]. In our analysis, a word is rare if it occurs once in our training corpus.

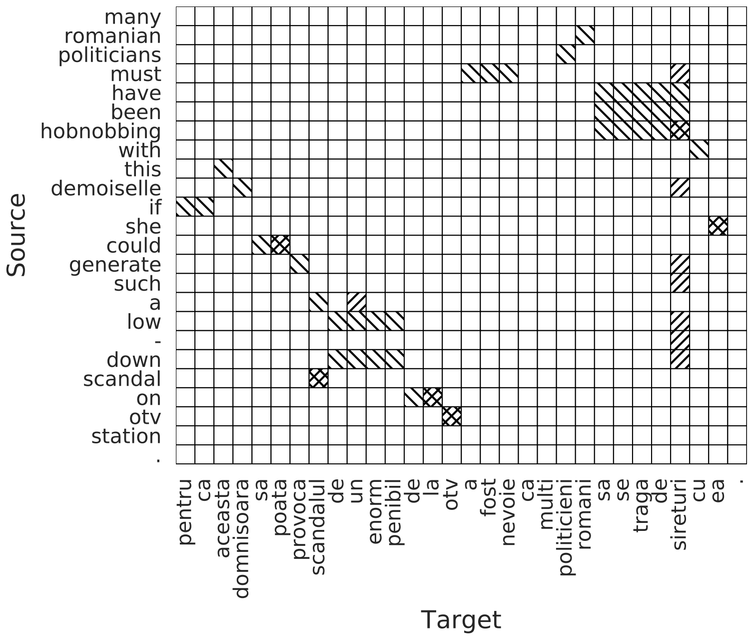


Figure 3.16: Example of alignment links for the Romanian rare word "sireturi". Back diagonal hatching, diagonal hatching and crossed diagonal hatching represent a reference alignment link, a predicted alignment link and a correctly predicted alignment link by `IBM-1 Giza++`. We can see that the word "sireturi" is erroneously linked to the English words "must", "demoiselle", "generate", "such", "low", "-" and "down".

To observe the garbage collector problem, we collect the number of source words linked to a rare target word. The fertility of rare words is the mean of these values. Basic statistics for rare words are in Table 3.9. For example, for the English-Czech language pair, the number of rare words in English is 461 and the number of aligned source words (and also the number of links) is the number in parentheses i.e. 558. The number of links in English-French and English-German is very small $\leq$ 40 links, which is not surprising because of their large size of the training corpus. We recognize also that two English words align often with one rare German word. This could be explained by a large number of many-to-one links (2 769 links in Table 3.8). The opposite trends are observed in the two Asian languages: one rare English word is often aligned to two Japanese/Vietnamese words (more than 11K and 55K links fall to the type one-to-many).

Complete results for our baselines are in [Ngo Ho, 2021, Appendix A.7]. Table 3.10 displays the scores of our baselines for English-Czech. In the reference data, there are 461 English rare words (558 links) and 1176 Czech rare words (1724 links). We recognize the largest effect of

Test corpus	Number of rare words		Fertility of rare words	
	English	Foreign	English	Foreign
English-French	11 (15)	23 (37)	1.4	1.6
English-German	6 (6)	19 (40)	1.0	2.1
English-Romanian	13 (21)	55 (88)	1.6	1.6
English-Czech	461 (558)	1 176 (1 724)	1.2	1.5
English-Japanese	171 (310)	100 (136)	1.8	1.3
English-Vietnamese	902 (1 751)	415 (419)	1.9	1.0

Table 3.9: Basic statistics for rare words in the test corpora

garbage collector on `IBM-1` (fertility of 4.25 for English with 1961 links and 2.86 for Czech with 3365 links) because of its simple structure based mainly on word co-occurrences. We notice that `Fastalign` provides the best remedy for this problem with the smallest fertility and the highest accuracy, higher scores than `IBM-4` with an explicit fertility model[21]. Note that ACC, F-score, Precision and Recall are computed for links involving rare target words. An observation is that in comparison to `Fastalign`, `Giza++ IBM-4` model significantly decreases the number of alignment links (Figure 3.5) but still keep many links for rare words (1468 links of `IBM-4` vs 700 links of `Fastalign`), which explains the higher recall. The lower precision can be attributed to GCP. Similar trends are found in other corpora.

Models	English						Foreign					
	#	FE	ACC	PRE	REC	F1	#	FE	ACC	PRE	REC	F1
`IBM-1 Giza++`	1961	4.25	85.54	15.96	56.09	24.85	3365	2.86	90.68	23.6	46.06	31.2
`Fastalign`	700	1.52	95.94	51.86	65.05	57.71	1489	1.27	95.84	55.41	47.85	51.35
`HMM Giza++`	1623	3.52	89.42	24.52	71.33	36.5	2878	2.45	93.61	38.26	63.86	47.85
`IBM-4 Giza++`	1468	3.18	90.83	28.13	74.01	40.77	2430	2.07	95	46.79	65.95	54.74

Table 3.10: Baselines for English-Czech: # links, fertility (FE), accuracy (ACC), F-score (F1), precision (PRE) and recall (REC) for the rare target words in the direction Czech-English and in the direction English-Czech

In order to check if a rare word is often longer than a frequent word, we observe word lengths (number of characters in a word) as a function of word occurrences [Powers, 1998]. Complete results for this analysis are in [Ngo Ho, 2021, Appendix A.6]. As can be seen in Figure 3.17, less frequent words have longer word lengths. Similar trends are found for other language pairs. For sub-word tokenization, this means that a rare word often decomposes into a long sequence of units.

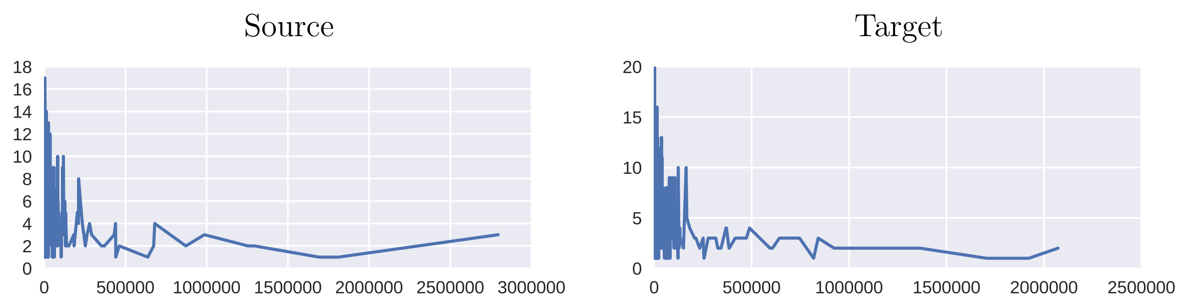


Figure 3.17: English-French: Word length as a function of word occurrence.

[21] This again confirms the finding of Dyer et al. [2013] that the reparameterization of `IBM Model 2` is a compelling replacement for the `Model 4`

3.7 How to process unknown words ?

Even more than rare words, aligning unknown words is always a difficult task. We would like to understand the behavior of models in predicting alignment links for this type of words.

In our analysis, a word is unknown if it does not appear in the training corpus. To observe the alignment patterns for unknown words, we apply the same method as for rare words. In fact, we collect the number of source words linked to unknown target words. The fertility of unknown words is the mean of these values. Basic statistics for unknown words are in Table 3.11. For instance, in English-French, the number of unknown target words in English is 157 and the number of aligned source words (and also the number of links) is the number in parentheses i.e. 294.

We recognize similar patterns for rare words as for unknown words. There is a difference in the case of English-French: the fertility of unknown words in English is larger than the corresponding count in French, which is explained by a large number of one-to-many and many-to-many links (Table 3.8).

Test corpus	Number of unknown words		Fertility of unknown word	
	English	Foreign	English	Foreign
English-French	157 (294)	64 (101)	1.9	1.6
English-German	15 (22)	58 (129)	1.5	2.2
English-Romanian	36 (49)	62 (96)	1.4	1.5
English-Czech	1 599 (2105)	2 546 (3 627)	1.3	1.4
English-Japanese	560 (1189)	240 (317)	2.1	1.3
English-Vietnamese	4 855 (5 959)	2 818 (1 902)	1.2	0.7

Table 3.11: Basic statistics for unknown words in the test corpora

We observe how the baselines process these unknown words in [Ngo Ho, 2021, Appendix A.8]. Recall that we concatenate training and test data in the previous experiments, which implies that there is no unknown word. Therefore, we replacing unknown words with a special token "UNK". Note that this token does not play the same role of rare words and the baselines have to learn the behavior of this special token. We also observe the behavior of these words in the case of concatenating training and test data. They act like rare words that happen at least once in training-test corpus.

Table 3.12 displays the scores for English-Czech. Note that we only report unknown words in the target side. We see that in the case of concatenating training and test corpus, these words and the rare words unsurprisingly share similar behaviors. The first observation, for the case of replacing unknown words with the UNK, is that `Fastalign` obtains a loss in F-score (in both directions) except for the direction German-English. Several observations can be made for the `Giza++` models:

- For the language pairs in large data condition (i.e., German and French), using the UNK token gives better F-score in both directions. This suggest that this token can help to overcome the problem of very rare words (happening at least once in training-test corpus) by reducing the effect of GCP.

- For small data condition, replacing unknown target words (English words in the direction Czech-English) with the UNK also helps all `Giza++` to outperform their counterparts (better F-scores). Note that this improvement comes from a large gain in precision and a small loss in recall. This behavior is found for the directions where the target side has the smaller number of unknown words than the source side i.e, the directions Czech-English, Romanian-English, English-Japanese and English-Vietnamese.

- An opposite behavior is that this UNK token in the target side makes a very large loss in recall, leading to a worse F-score. This also suggest that this special token often aligns with the NULL token. We can see this tendency in the directions where the target side has the larger number of unknown words than the source side i.e., the direction English-Czech, English-Romanian, Japanese-English and Vietnamese-English.

Models	English						Foreign					
	#	FE	ACC	PRE	REC	F1	#	FE	ACC	PRE	REC	F1
	Concatenation											
IBM-1 Giza++	6931	4.33	85.4	16.87	55.53	25.87	8487	3.33	89.86	20.9	48.91	29.29
Fastalign	2118	1.32	96.29	59.54	59.9	59.72	3056	1.2	96.22	57.04	48.06	52.16
HMM Giza++	5702	3.57	89.75	27.24	73.78	39.78	7488	2.94	92.59	32.41	66.91	43.67
IBM-4 Giza++	5132	3.21	91.11	30.79	75.06	43.66	6058	2.38	94.39	40.82	68.18	51.07
	Replacing unknown words by the token UNK											
IBM-1 Giza++	2124	1.33	93.18	25.94	26.18	26.06	2077	0.82	94.5	25.37	14.53	18.48
Fastalign	2076	1.3	95.2	47.69	47.03	47.36	2820	1.11	95.39	45.25	35.18	39.58
HMM Giza++	1854	1.16	95.37	49.51	43.61	46.38	1869	0.73	95.88	53.93	27.79	36.68
IBM-4 Giza++	1977	1.24	95.1	46.38	43.56	44.93	1839	0.72	95.71	50.19	25.45	33.77

Table 3.12: Baselines for English-Czech: # links, fertility (FE), accuracy (ACC), F-score (F1), precision (PRE) and recall (REC) for the unknown target words in Czech-English and in English-Czech.

Handling unknown words The two well-used techniques to handle unknown words are subword tokenization (e.g., BPE; Section 2.3.2) and character-based models. Since the smallest unit is a character, these models clearly help to solve unknown words, especially for morphologically rich languages. Note that we do not extract character-level alignment but encode a sequence of characters to obtain a word representation, which means that we keep the word boundary. For example, a sentence "it was a fine morning ." becomes "[i,t], [w,a,s], [a], [f,i,n,e], [m,o,r,n,i,n,g], [.]". Character-based representation level is mainly used to improve or replace the word embedding [O'Neill and Bollegala, 2018]. The application of character-based representation can be found in language modeling [Kang et al., 2011, Kim et al., 2015, Costa-jussà and Fonollosa, 2016, Labeau and Allauzen, 2017, Nicolai et al., 2018, Renduchintala et al., 2018]. Chung et al. [2016] remove the restriction of word boundaries to obtain a character-level decoder and Lee et al. [2017] extend it to a fully character-level model. Cherry et al. [2018] underline the higher performance of character-level models compared with subword-level models if they are given enough model capacity. The effectiveness has been demonstrated in other domains such as word alignment [McCoy and Frank, 2018] and sentence pair modeling [Lan and Xu, 2018]. An important difference between BPE and character-based representation is that the latter only allows training representations for unknown words.

3.8 Are function words harder to align than content words ?

Each language has a different way to express a grammatical or structural relationship with other words, often taking the form of so-called function words. The alignment task for function words mainly depends on annotators. For example, in the sentence pair ("Les armes de les soldats ", "The soldier weapons"), the French word "de" remains unaligned or aligns with the punctuation ",". There were several attempts at providing an annotation style guide e.g. English-French[22], English-Czech [Kruijff-Korbayová et al., 2006], Hindi-English [Gupta and

[22] https://nlp.cs.nyu.edu/blinker/

Yadav, 2010], Spanish-English [Lambert et al., 2005], English-Swedish [Ahrenberg, 2007] etc, each containing detailed procedure to handle such cases.

To observe how models process these function words, we categorize words into two groups based on their PoS: content words include nouns, verbs, adjectives and adverbs and function words for the remaining PoS. To obtain PoS in our analysis, we use Spacy [23] for English, French, German, Japanese and Romanian; VnTagger[24] [Le-Hong et al., 2010] for Vietnamese and RACAI [Dumitrescu et al., 2017] for Czech. Note that each tool uses a different annotation system, we hence transform them into Universal POS tags [25].

Basic statistics for content words and function words are in Table 3.13 and Table 3.14 respectively. We recognize that the difference between the number of content words and function words in both English and foreign languages is small, except the case of Czech with about 5 000 and 10 000 words (because of the large size of testing data). Some observations can be made:

- The number of aligned content words in English is larger than their foreign counterparts (French and German). We see an opposite trend for function words.
- We observe a different situation for Romanian, Japanese and Vietnamese, the number of aligned content English words is smaller than their foreign counterparts and an opposite trend for function words. Note that the difference is significantly larger in Japanese (content words) and Vietnamese (content and function words).
- In English-Japanese and English-Vietnamese, about half of the function words are unaligned words. As can be seen in Figure 3.3 (page 50), the function words "to", "a" and "of" are unaligned.
- In Vietnamese, the number of content words is substantially larger than the number of function words and only a small portion of function words is aligned. This highlights the prevalence of content words in alignment.
- In the case of Czech, the number of English words in both grammatical classes is greater than the word numbers in Czech. This is expected given the amount of many-to-one links. Moreover, the number of content words is larger than the number of function words.

Test corpus	English			Foreign		
	# words	# aligned words	# links	# words	# aligned words	# links
English-French	3 646	3 498	10 458	3 268	3 165	7 968
English-German	5 818	5 440	6 184	4 359	4 037	5 349
English-Romanian	2 917	2 695	3 441	2 988	2 809	3 709
English-Czech	32 727	31 335	38 445	31 355	29 149	42 326
English-Japanese	8 801	7 988	12 607	16 022	15 050	18 560
English-Vietnamese	26 993	24 887	49 191	79 433	71 988	74 573

Table 3.13: Basic statistics of content words for the test corpora

To observe the behaviors of our baselines, we count the number of correct/incorrect alignment links [Ngo Ho, 2021, Appendix A.9.1] and also the unaligned words [Ngo Ho, 2021, Appendix A.9.2] for our two PoS categories. Figure 3.18 displays alignment links for English-Czech. We recognize that `Fastalign` improves content words with a simple assumption about

[23] https://spacy.io/
[24] https://vlsp.org.vn/wiki/tools
[25] https://universaldependencies.org/u/pos/

Test corpus	English			Foreign		
	# words	# aligned words	# links	# words	# aligned words	# links
English-French	3 374	3 195	6 980	4 493	4 247	9 470
English-German	4 700	4 115	4 349	5 600	4 636	5 184
English-Romanian	2 538	2 253	2 547	2 327	2 015	2 279
English-Czech	27 354	25 063	28 978	21 526	19 662	25 097
English-Japanese	22 021	14 869	20 770	18 381	13 001	14 817
English-Vietnamese	43 056	22 795	32 544	15 320	6 980	7 162

Table 3.14: Basic statistics of function words for the test corpora

the distortion model, which is in some cases better than `IBM-4`. This strength of `Fastalign` can be observed in other language pairs/directions.

Note that in Section 3.3, we showed that `IBM-4` did not generate more correct links than the other models, but simply removed incorrect alignment links (source words are aligned to NULL token). Function words seem to mostly benefit from this reduction e.g., FP of function words decreases (Figure 3.18). However, for the reference alignments, most function source words must be aligned, which yields a large number of incorrect unaligned source words (Figure 3.19). Similar trends are also observed in other models and in both directions. These behaviors require a model that encodes the necessary information for function words, especially a model for NULL token.

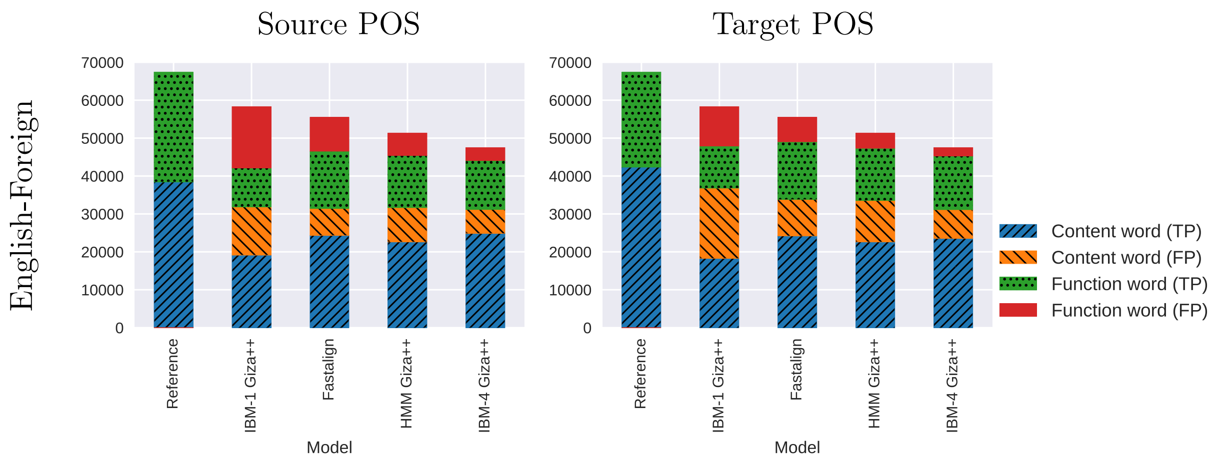


Figure 3.18: Baselines for English-Czech: The number of target words that align with a content/function source word (left graph). The number of source words that align with a content/function target words (right graph).

Figure 3.19: Baselines for English-Czech: The number of unaligned content/function source word (left graph). The number of unaligned content/function target words (right graph).

3.9 Improvements by symmetrization and agreement

We study symmetrical alignment by considering two methods: intersection and grow-diag-final (GDF) (Section 2.4.3.1). The intersection method helps to evaluate the agreement between asymmetrical alignments. Symmetricized results for our baselines are in [Ngo Ho, 2021, Appendix A.10.1] and [Ngo Ho, 2021, Appendix A.10.2]. Table 3.15 shows the statistics for intersection alignments in the case of English-Czech. We recognize that more complex models achieve higher levels of agreement (ratio on both directions). Note that `Fastalign` improves this ratio more than `HMM` and `IBM-4` in the case of Czech, Japanese and Vietnamese.

Using GDF, the performance of our baselines is improved. For example, `IBM-1` gains about -10 AER and the more complex baselines achieve -2/3 AER. Recall that the reference alignments are symmetrical. Therefore, symmetrization is always a method to improve the alignment performance.

Models	# links	Ratio		AER	F1	PRE	REC	ACC	FE	
		En-XX	XX-En						En	Fr
`IBM-1 Giza++`	23298	0.45	0.4	40.22	45.12	87.85	30.36	97	0.39	0.55
`Fastalign`	36091	0.71	0.65	20.68	63.06	90.43	48.41	97.7	0.63	0.77
`HMM Giza++`	28415	0.6	0.55	25.65	57.2	96.46	40.65	97.53	0.51	0.68
`IBM-4 Giza++`	30648	0.69	0.65	21.43	60.84	97.33	44.24	97.69	0.58	0.73

Table 3.15: Intersection alignment: The number of alignment links, their ratio to the total number of alignment links predicted by the model, alignment error rate (AER), accuracy (ACC), F-score (F1), precision (PRE), recall (REC) and average fertility (FE) for English-Czech

Models	English-Foreign		Foreign-English		GDF			
	AER	F1	AER	F1	AER	F1	PRE	REC
`IBM-1 Giza++`	45.09	46.75	48.47	42.88	35.47	52.67	71.16	41.81
`Fastalign`	25.75	64.09	25.3	62.86	23.3	66.93	72.95	61.82
`HMM Giza++`	27.86	61.22	30.38	57.28	25.25	62.96	75.67	53.91
`IBM-4 Giza++`	20.92	65.7	26.5	59.81	19.13	66.67	84.22	55.17

Table 3.16: Grow-diag-final: Alignment error rate (AER), F-score (F1) for English-Czech

3.10 Do sentence lengths shape alignment patterns ?

We study sentence lengths (in words) [Ngo Ho, 2021, Appendix A.11.1], by observing the difference between the length of a source and a target sentence. This value is computed by subtracting the length of the foreign language sentence from the length of the English sentence, shown in [Ngo Ho, 2021, Appendix A.11.2].

All of the sentences in the English-German/French test corpus are short (< 50 words) whereas the length of some sentences in the other corpora is as large as 100 words. Figure 3.20 shows that the length difference in the training set could be large (≥ 100 words), created by a small number of sentences, except for French and Romanian sentences. As expected, a high density of sentences appears around the difference value 0. English-French and English-German test sets (3.21) bring out two opposite patterns:

- The high density of length difference bends left, meaning that the length of foreign (French) sentences is often greater than the corresponding English sentence.

- The high density bends right in the case of English-German, showing the opposite trend.

This issue has direct impacts on the number of unaligned words in both sides and the type of alignment, specially in the case of the asymmetrical alignment models. For example, in the first pattern, we could observe two trends: one English word is often aligned to many foreign words and/or there is a large number of unaligned foreign words.

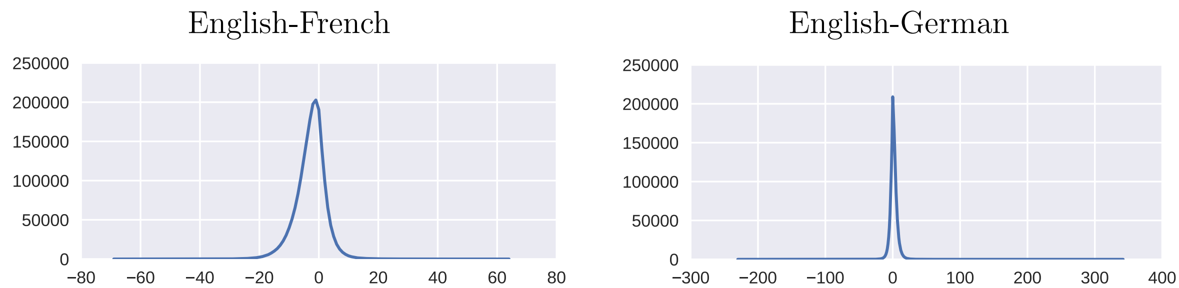


Figure 3.20: Length differences in English-French and English-German training sets. The axis x shows the length difference values while y represents the number of sentences.

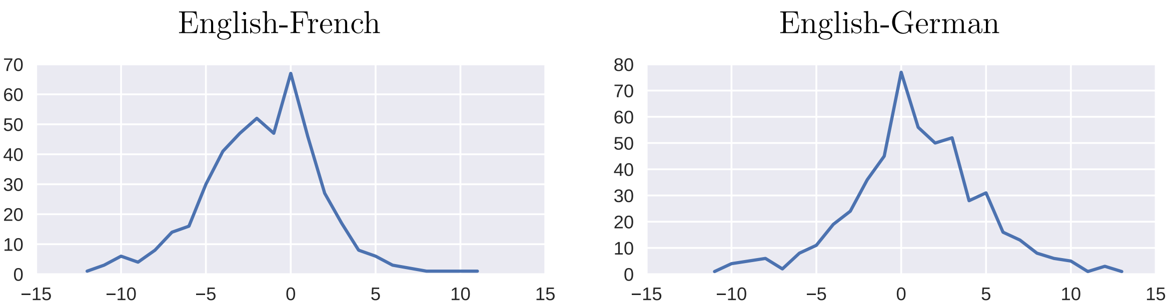


Figure 3.21: Length differences in English-French and English-German testing sets. The axis x shows the length difference values while y represents the number of sentences.

We observe AER scores as a function of sentence length difference (i.e., subtracting the length of the target sentence from the length of the source sentence), shown in [Ngo Ho, 2021, Appendix A.11.6]. An observation is that smaller length differences often obtain better AER scores as can be seen in Figure 3.22.

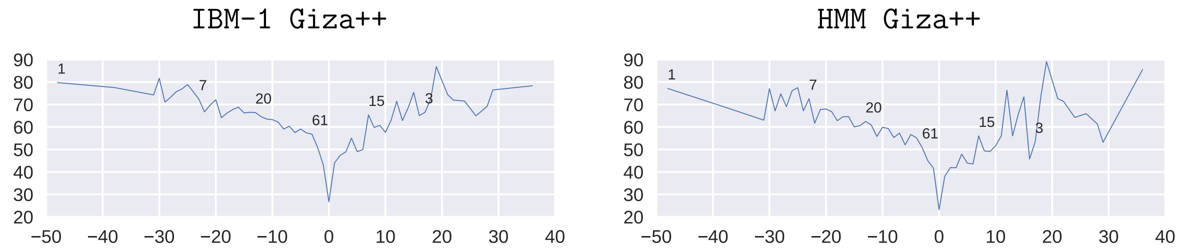


Figure 3.22: `IBM-1` and `HMM Giza++` for the direction English-Japanese: AER score as a function of sentence length difference. The x-axis shows the sentence length difference. The y-axis represents the AER. The annotation displays the number of sentences.

The longer the sentence, the harder the prediction. We observe AER scores as a function of sentence length on both sides, shown in [Ngo Ho, 2021, Appendix A.11.5]. For our baselines, the longer the sentences are harder for alignment prediction, e.g., the case of English-Czech in Figure 3.23. We see that the scores of `IBM-4` fluctuate around 0.2 for almost sentences with length less than 40, followed by a rise from about 0.3 to 0.6 for the rest of the sentences. This situation is also observed for other languages.

One obvious reason for this problem is that longer sentences provide more alignment alternatives which also increase the chance of producing alignment errors. Another reason is from rare/unknown words: longer sentences often include more rare/unknown words. To observe this, we plot the average number of unknown/rare words as a function of sentence length, displayed in [Ngo Ho, 2021, Appendix A.11.3]. We recognize that there are more unknown/rare words in longer sentences e.g., Czech words in Figure 3.24. This clearly worsens ”garbage collector problem". Therefore, one obvious solution for long sentences is to improve the prediction for unknown/rare words.

In addition, word repetition happens more often in longer sentences, which is also harmful to the performance. As an illustration in Figure 3.25, the English word "shall" repeats twice in the English sentence with a length equal to 64 and incorrectly aligns with Czech unknown word "písm". This is a likely sign of a too confident translation model, requiring a better distortion model for long sentences. To observe the prevalence of word repetition, we plot the average number of words that repeat at least twice as a function of sentence length, displayed in [Ngo Ho, 2021, Appendix A.11.4]. We see that the repetition of both English and Czech words is clearer for longer sentences e.g., Figure 3.26.

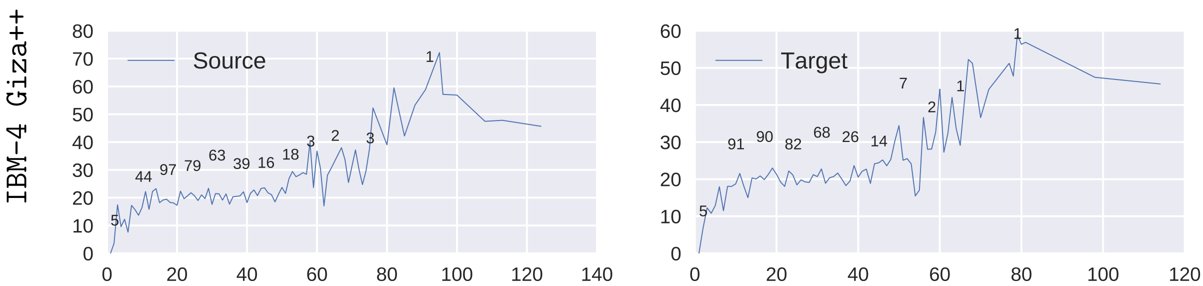


Figure 3.23: The direction English-Czech: AER score for `IBM-4 Giza++` as a function of sentence length. The x-axis shows the sentence length. The y-axis represents the AER. The annotation displays the number of sentences.

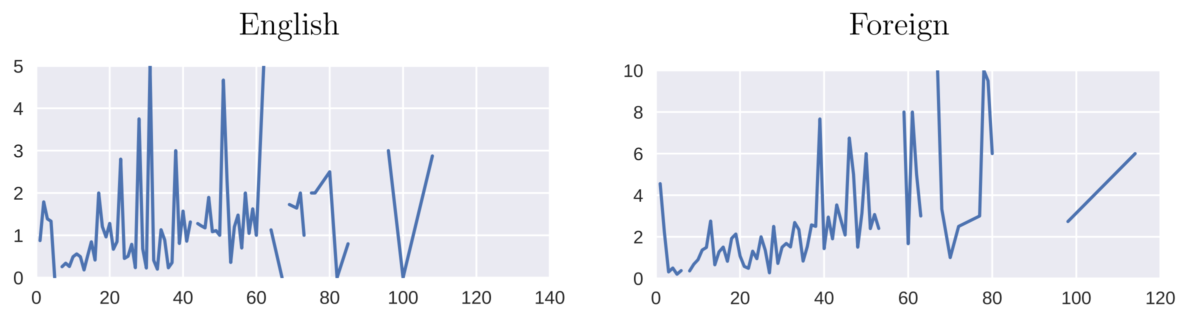


Figure 3.24: Number of unknown/rare words as a function of sentence length for English-Czech

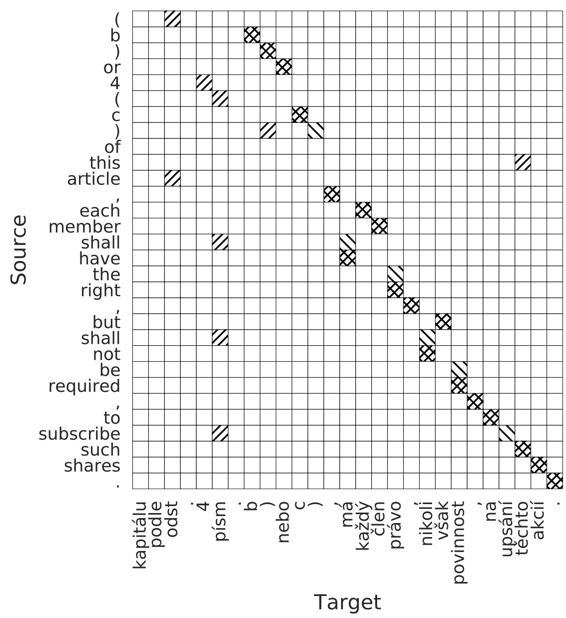


Figure 3.25: Example of word repetitions in a long source sentence (64 words): Only a part of this sentence is displayed. Back diagonal hatching, diagonal hatching and crossed diagonal hatching represent a reference alignment link, a predicted alignment link and a correctly predicted alignment link by `Fastalign`. English word "shall" repeats twice and incorrectly aligns with Czech unknown word "písm".

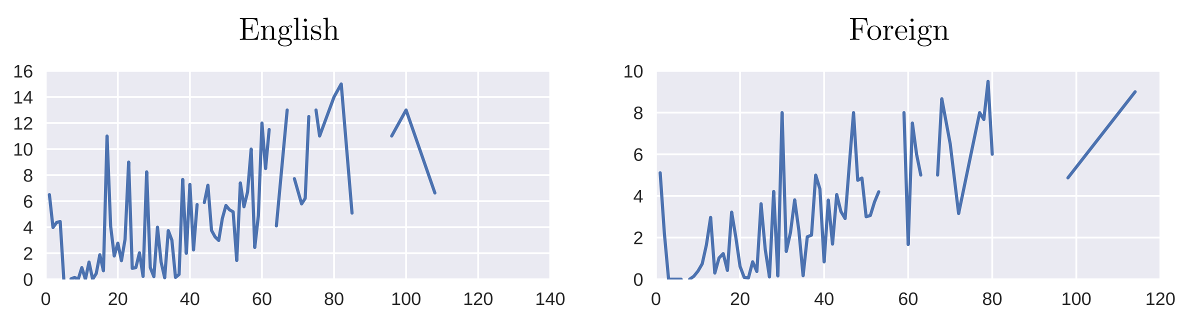


Figure 3.26: Number of words that repeat at least twice as a function of sentence length for English-Czech

3.11 Summary

In this chapter, we introduced a list of evaluation methods and reported results on six corpora English with French, German, Romanian, Czech, Japanese, and Vietnamese. We presented basic statistics for training (Section 3.1.1) and test corpora (Section 3.1.2) including the number of sentences, number of words and vocabulary size. We discussed that the human reference alignments (sure/possible links) introduced a bias for the AER score, a common method to measure model performance (Section 3.2). This highlighted that these sure and possible links need to be observed under different perspectives. We showed effects of sentence length, which has a strong impact on alignment patterns. It is clear that the baselines do not well predict alignment links for long sentences. The first observation that motivates all the rest: the problem is far from solved especially for distant language pairs and/or in low resource conditions. Even German/English (high resource and same family) the alignment scores are quite bad. Another consideration was about unaligned words (Section 3.3), which exhibit an undesirable behavior: the distortion model does not help to generate more correct links but simply removes incorrect links, creating more incorrectly unaligned words. Moreover, predicting correct jumps is still a difficult task for our baselines because of simplistic underlying assumptions and a lack of context information (Section 3.5). Other problems taken into account are the garbage collector problem for rare words (Section 3.6) and also the function word problem (Section 3.8). In fact, function words are too often aligned to the NULL token. These problems come from the word co-occurrence approach that underlies statistical models. Symmetrical alignment remains an important line of research for corpora including a large number of many-to-many links (Section 3.4). In addition, the rise of the agreement level is also a challenge to improve our baselines. We summarize some of our findings as follows:

- English-French: With a large number of training parallel sentences, the problem of rare/unknown words seems less relevant. The models which can generate many-to-many links, benefit from its large number of many-to-many links, and also possible reference links. With these possible links (76.8%), a low recall for aligned words less significantly impacts the AER. Moreover, English and French share similar grammar structures e.g., SVO. This can make the alignment task simpler for this language pair than for other pairs.

- English-German: This language pair is also in large data condition with a small number of unknown/rare words. Asymmetrical models can still work for this language pair because of a large number of one-to-one links (6000 links accounting for about 60% of the links). We see difficulties for unaligned words when there are about 900 alignments to the NULL token in the English side and a high ratio (13%) of unaligned German words.

- English-Czech: We use the training corpus in small data condition and there is a large number of sentence pairs in the test corpus. This help to explore a problem for unknown/rare words. In the direction English-Czech, asymmetrical models can better benefit from many-to-one links with 30% of the total. The test corpus contains ~23K possible reference links (34.3%) that help to reduce the impact of a low recall on the AER.

- English-Romanian: We also use a small data for this language pair but it does not make the alignment task more difficult for unknown/rare words. Asymmetrical models seem fine for this pair with an even distribution of alignment types. There is no possible reference link, which means that a low recall for aligned words (a large number of NULL links) directly impacts the AER.

- English-Japanese: As we consider a small data condition, the problem of unknown/rare words creates a significant issue. Note that both English and Japanese have a high ratio of unaligned words (respectively 23.6% and 18.1%). Asymmetrical models can take advantage of ~11K many-to-one links (35%) in the direction Japanese-English. However, different word orders between English and Japanese (e.g., SVO and SOV) create a strong obstacle for the word alignment task. Moreover, the test set only contains sure links, yielding that a large number of NULL links can greatly affect the AER.

- English-Vietnamese: We see similar problems for unknown/rare words because of its small number of training sentence pairs and of its large test size. It shares the same problem of Romanian and Japanese where there is no possible reference link. Asymmetrical models in the direction Vietnamese-English outperform their counterparts in the opposite direction due to a large proportion of many-to-one links, namely 67.6%. In addition, the high ratios of unaligned English and Vietnamese words are difficult challenges for NULL models.

Our analyses are based on the set of human reference alignments and these alignments mainly depend on the perception of annotators. Therefore, we stress that alignment evaluation is a complex and difficult task. In addition, we highlight that it always requires good guidelines for annotators. Different guidelines can yield important changes for sure/possible links, alignment types, unaligned words, function words, and also word orders.

Even though the performance of statistical generative alignment models seems fair for related languages (e.g., English-French), there is still much room for improving automatic alignments produced by standard tools such as `Giza++` or `Fastalign`. Under the dawn of neural network architectures, we will discuss how to apply neural networks for the word alignment task in the next chapter and we try to see how much neural models can help to solve the above-mentioned challenges.

Chapter 4

Neural word alignment models

Until recently, the most successful alignment models were statistical, as represented by the IBM Models [Brown et al., 1993b] and the HMM model [Vogel et al., 1996]. These models use unsupervised estimation techniques to build alignment links at the word level, relying on large collections of parallel sentences. Such approaches are typically challenged by low-frequency words, whose cooccurrences are poorly estimated and they also fail to take into account context information in alignment. Even though their performance seems fair for related languages (e.g. French-English), these was amply confirmed by our analysis of Chapter 3.

As is the case for most NLP applications [Collobert et al., 2011], and notably for machine translation (MT) [Cho et al., 2014a, Bahdanau et al., 2015, Luong et al., 2015], neural-based approaches offer new ways to address some of these issues. One important reason for this success is the implicit feature extraction performed by neural networks, which represent each word as a dense low-dimensional vector and effectively extend word representations by vector concatenation [Young et al., 2017]. Following up on the work of Yang et al. [2013], Tamura et al. [2014], Alkhouli et al. [2016], Wang et al. [2017, 2018], we focus here on neural word alignments, trying to precisely assess the benefits of neuralizing standard word alignment models. We thus design and implement multiple neural variants of the IBM and HMM models. We not only report improved AER scores but also detail the positive impact of these neural baselines on major alignment error types such as aligned and non-aligned words, rare vs frequent words, etc (Chapter 3). We also discuss the relevance of our neural network variants for each language pair and error type. Therefore, we make the following contribution:

- A systematic comparison of several neural models for word alignments including context-independent models, contextual models, and character-based models, which allow us to establish strong baselines for further studies.

- Our experiments notably reveal that neuralized versions of standard alignment models vastly outperform their discrete counterparts, but also show that there still exists much room for improvements, especially when dealing with morphologically rich languages or in low-resource settings.

In this chapter, we first present an overview of neural networks and several architectures used in NLP in Section 4.1. In Section 4.2, we quickly survey related works for neural word alignment. We then describe our contributions: (a) neuralizations of the translation models in Section 4.3; (b) neuralizations of the distortion models in Section 4.4. We give details of our training algorithm (Section 4.5) and our experiments (Section 4.6). We finally discuss our alignment results in Section 4.7 where we present the alignment errors that are fixed and those that still challenge statistical and neural models. A shorter version of this work is published in Ngo-Ho and Yvon [2019].

Contents

4.1 Artificial neural networks in NLP

This section describes artificial neural networks and discusses several applications of neural methods in NLP [Koehn, 2010, Cho, 2014]. We refer to Goodfellow et al. [2016] and Young et al. [2017] for a thorough introduction to the field. An artificial neural network (NN) consists of multiple neurons (units) and multiple layers of neurons. Information flows through these layers from an input layer, through one or several hidden layers and to an output layer. The result of each layer can be considered as a representation of data.

Activation functions: Each unit of a layer receives information from units of the previous layer by computing the weighted sum of the input values. They control the outputs of a layer (i.e., decide if a neuron can be fired or not) by producing the activation values [Nwankpa et al., 2018]. Some common activation functions are:

- Linear function means an affine transformation. In our work, a layer using this function often helps to modify the size of a data representation.

$$f(x) = ax \tag{4.1}$$

- Hyperbolic tangent activation function: This function is similar to the identity function near 0. This means that training a neural network with this function resembles training a linear model if the activations of this network can be kept small, which makes this training easier [Goodfellow et al., 2016]. This activation function is often used in our models.

$$f(x) = \tanh(x) = \frac{2}{1+e^{-2x}} - 1 \tag{4.2}$$

- Softmax function: It is used for the output layer since it helps to represent a probability distribution over a discrete variable with multiple classes e.g. vocabulary.

$$f(x_j) = \frac{\exp x_j}{\sum_{j'=1}^{j'=J} \exp x_{j'}} \quad \forall j \in [1, J] \tag{4.3}$$

- Softplus function: It helps to generate non-negative value.

$$f(x) = \log(1 + \exp(x)) \tag{4.4}$$

Learning algorithm: Gradient descent algorithm [Curry, 1944], an optimization algorithm, is commonly used in neural networks. This algorithm minimizes an objective function $J(\theta)$ with parameters $\theta \in \mathbb{R}^d$. It updates the parameters in the opposite direction of the gradient of the objective function $\nabla_\theta J(\theta)$. Mini-batch gradient descent performs a parameter update for K sentence pairs $(\mathbf{f}, \mathbf{e})_1^K$. The model parameters at step t can be computed as:

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta J(\theta_t, (\mathbf{f}, \mathbf{e})_1^K)) \tag{4.5}$$

where η is the learning rate determining the size of the steps to reach a minimum. One issue of the vanilla mini-batch gradient descent is how to select an appropriate learning rate at each mini step. Another issue of minimizing highly non-convex error functions that are typically used for neural networks is avoiding getting trapped in their numerous suboptimal local minima. Therefore, several algorithms are proposed to deal with the learning rate, which means that they compute adaptive learning rates for each parameter. This is for instance the case of Adagrad [Duchi et al., 2011], Adadelta [Zeiler, 2012], RMSprop [1]), Adam [Kingma and Ba, 2014], etc. Note that RMSprop, Adadelta, and Adam are very similar algorithms. However Kingma and Ba [2014] show that its bias-correction helps Adam to slightly outperform RMSprop towards the end of optimization as gradients become sparser. Therefore, we use Adam as our learning algorithm. Adam stores an exponentially decaying average of past squared gradients and keeps

[1] https://keras.io/api/optimizers/rmsprop/

an exponentially decaying average of past gradients m_t.

$$g_t = \nabla_\theta J(\theta_t, (\mathbf{f}, \mathbf{e})_1^K)) \tag{4.6}$$
$$m_t = \beta_1 m_{t-1} + (1 + \beta_1) g_t \tag{4.7}$$
$$v_t = \beta_2 v_{t-1} + (1 + \beta_2) g_t^2 \tag{4.8}$$
$$\hat{m_t} = \frac{m_t}{1 - \beta_1^t} \tag{4.9}$$
$$\hat{v_t} = \frac{v_t}{1 - \beta_2^t} \tag{4.10}$$
$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v_t}} + \epsilon} \hat{m_t} \tag{4.11}$$

where m_t is the decaying averages of past gradients, v_t is the decaying averages of past squared gradients, $\hat{m_t}$ and $\hat{v_t}$ are bias-corrected first and second moment estimates. We refer to Ruder [2017] for an overview of gradient descent algorithms.

4.1.1 Word embeddings

One significant drawback of shallow machine learning models (e.g., SVM or logistic regression) in NLP is the curse of dimensionality because linguistic information typically is represented with very high dimensional and sparse features. Neural networks based on word embeddings, low dimensional, and distributed representations, achieve better results on various NLP tasks. Collobert et al. [2011] suggest that a simple multilayer neural network architecture could solve with high accuracy a host of NLP tasks such as named-entity recognition, semantic role labeling, and POS tagging. Word embeddings are based on the distributional hypothesis: words sharing similar context have similar meaning. In other words, word embedding can capture syntactical and semantic information based on its context [Young et al., 2017]. [Bengio et al., 2003] use distributed word representations in a language model, turning n-grams distributions into smooth functions of the word representations. Mikolov et al. [2013] propose the CBOW and skip-gram models where they construct word embeddings based on the surrounding context words.

A word w is considered as an index i in a finite dictionary of size V. It is represented by a one-hot encoded vector v in a high-dimensional discrete space $\mathbb{R}^V$. All values of v are null except the value at the position i equal to 1. We observe a simplified version of the CBOW model where only one word is considered in the context, displayed in Figure 4.1. In the process of predicting the target word, CBOW learns its representation. The input is a one-hot encoded vector of size V. The hidden layer contains N neurons. The output is passed to the softmax function that computes a distribution over all V words in the vocabulary and the highest value in the output vector indicates the output word. The layers are connected by weight matrices $W_1 \in \mathbb{R}^{V \times N}$ and $W_2 \in \mathbb{R}^{N \times V}$.

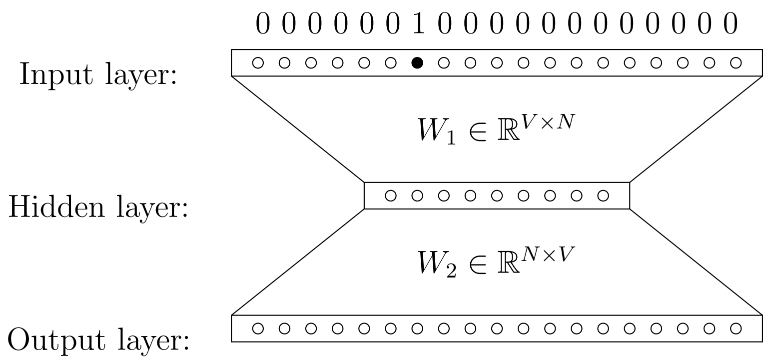


Figure 4.1: Simplified version of the CBOW with only one word in context.

4.1.2 Convolutional neural networks (CNN)

A CNN extracts higher-level features from constituting words or n-grams [Young et al., 2017]. The first application of CNN-based frameworks for NLP tasks is found in the works of Collobert and Weston [2008] where a word embedding is constructed via a look-up table. In [Zhang and LeCun, 2015], CNNs can show their usefulness for text understanding without the knowledge of words, phrases, sentences and any other syntactic or semantic structures with regards to a human language. Several application of CNNs are text classification [Kim, 2014], semantic parsing [Yih et al., 2015], paraphrase detection [Bogdanova et al., 2015], speech recognition [Abdel-Hamid et al., 2014], machine translation [Renduchintala et al., 2018, Gehring et al., 2017], etc.

CNNs process sentences as follows. Given a sentence e_1^I of I words, $E(e_i) \in \mathbb{R}^{1\times d}$ is the embedding of word e_i. This sentence can be represented as an embedding matrix $W \in \mathbb{R}^{N\times d}$. Let $w_{i:i+j}$ refer to the concatenation of vectors $w_i, w_{i+1}, ..., w_j$. The convolution operation, performed on this input embedding layer, includes a filter $k \in \mathbb{R}^{h\times d}$. This filter is applied to a window of h words to produce a new feature. As an illustration, a feature c_i is generated using a window of words $w_{i:i+h-1}$:

$$c_i = f(w_{i:i+h-1}k + b) \tag{4.12}$$

where $b \in R$ is the bias term and f is a non-linear activation function. The filter k is applied to all possible windows using the same weights to create the feature map $c = [c_1, c_2, ..., c_{N-h+1}]$. Note that CNN can contain a number of convolutional filters of different sizes. They slide over the entire word embedding matrix. Each filter extracts a specific pattern of n-gram. This is then followed by a max-pooling operation that applies a max operation on each filter to obtain a fixed-length output and reduce the dimensionality of the output.

4.1.3 Recurrent neural networks (RNN)

RNNs help to process a sequential information where they apply the same weight set recursively over each instance of the sequence: the output depends not only on the present inputs but also on the previous computation. This also means that RNNs have memory over previous instance of the sequence. These sequences are typically represented by a fixed-size vector of tokens which are fed sequentially (one by one) to a recurrent unit. This type is naturally suited for many NLP tasks such as language modeling [Mikolov et al., 2010, Mikolov et al., 2011, Sutskever et al., 2011], machine translation [Liu et al., 2014, Auli et al., 2013, Sutskever et al., 2014], speech recognition [Robinson et al., 1996, Graves et al., 2013, Graves and Jaitly, 2014, Sak et al., 2014] and also image captioning [Karpathy and Li, 2014].

In Figure 4.2, we observe a simple RNN network. In this network, x_i is the input to the network at time step i and h_i represents the hidden state at the same time step. h_i is computed based on the current input x_i and the previous time step's hidden state h_{i-1}:

$$h_i = f_1(W_1 x_i + W_2 h_{i-1} + b_1) \tag{4.13}$$
$$o_i = f_2(W_3 h_i + b_2) \tag{4.14}$$

where W accounts for weights that are shared across time, f_1 and f_2 are the activation functions, o_i is the output of the network, and b_1, b_2 are the bias terms. In the context of NLP, x_i could be a one-hot encoding or a word embedding.

Note that the gradient flow in simple RNNs often yields exploding and vanishing gradients which makes it difficult to learn and tune the parameters in the earlier layers for long sentences. Gradient clipping can solve the problem of exploding gradients by scaling a gradient if it is larger than a threshold. The vanishing gradient problem was overcome by various networks such as long short-term memory units (LSTM), gated recurrent units [Cho et al., 2014b] etc.

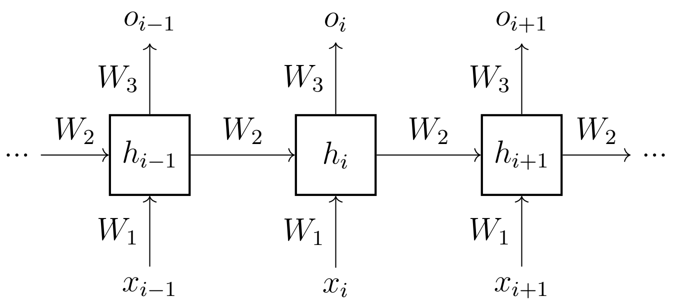


Figure 4.2: simple RNN network

Long short-term memory (LSTM) network is a particular case of RNN where it can control the memory for each instance of a sequence [Graves, 2013, Hochreiter and Schmidhuber, 1997, Gers et al., 2000]. LSTM contains several gates (e.g., input gate, forget gate, output gate) controlling how information is kept and forgot [Chung et al., 2014].

- The input gate regulates how the much new input changes the memory state.
- The forget gate regulates how much of the prior memory state is retained (or forgotten)
- The output gate regulates how strongly the memory state is passed on to the next layer.

$$\mathrm{fg}_i = \sigma(W_1 x_i + b_1) \tag{4.15}$$
$$\mathrm{ip}_i = \sigma(W_2 x_i + b_2) \tag{4.16}$$
$$\mathrm{op}_i = \sigma(W_3 x_i + b_3) \tag{4.17}$$
$$c_i = \mathrm{fg}_i \odot c_{i-1} + \mathrm{ip}_i \odot \tanh(W_4 x_i + b_4) \tag{4.18}$$
$$h_i = \mathrm{op}_i \odot \tanh(c_i) \tag{4.19}$$

where σ is the logistic sigmoid function; $\mathrm{fg}_i, \mathrm{ip}_i, \mathrm{op}_i$ are respectively a forget gate, input gate and output gate at moment i. c_i, a memory cell, is updated by partially forgetting the existing memory c_{i-1} and adding a new memory content $\tanh(W_4 x_i + b_4)$.

Bidirectional RNNs (BiRNN) are introduced by Schuster and Paliwal [1997]. A variant of this NN is the BiLSTM, presented in [Graves and Schmidhuber, 2005]. RNN is applied in both directions starting from the first state to the last state (forward mode) and from the last state to the first state (backward mode). We then concatenate the forward $\overrightarrow{h_i}$ and backward $\overleftarrow{h_i}$ hidden states to obtain the hidden state $h_i = [\overrightarrow{h_i}, \overleftarrow{h_i}]$ and feed it to the output layer. We replace the equation (4.14) by:

$$o_i = f_2(W_3 h_i + b_2) \tag{4.20}$$

4.1.4 Sequence-to-sequence models

The main application of a recurrent neural network is to model language as a sequential process. Given all previous words, such a model predicts the next word. After reaching the end of the sentence, the model predicts the translation of the sentence, one word at a time. A sequence-to-sequence model usually consists of an encoder and a decoder. The encoder transforms the source sentence into a higher dimensional vector. The decoder predicts the target sentence based on this vector. This architecture Encoder-Decoder is introduced by Sutskever et al. [2014] and Cho et al. [2014b]. It is mainly used in many NLP tasks such as question answering systems [Afrae et al., 2020, He et al., 2017], machine translation [Cho et al., 2014a, Bahdanau et al., 2015, Luong et al., 2015, Luong and Manning, 2015, Cheng et al., 2016, Yang et al., 2017, Cherry et al., 2018, Morishita et al., 2018, Wang et al., 2020], just to name a few.

4.1.4.1 Encoder-Decoder

Sequence-to-sequence models are designed to transform a source sequence f_1^J into a target sequence e_1^I. The source sentence f_1^J consists of a sequence of J tokens $(f_1, \ldots, f_J)$ and the target sentence e_1^I consists of I tokens $(e_1, \ldots, e_I)$. In detail, the task of the encoder is to provide a representation of the source sentence: (a) Encode source sequence of J word embeddings $(E(f_1) \ldots E(f_J))$. (b) Generate a sequence of hidden states h_1^J. (c) Produce a dense representation c of this sentence (the source sentence embedding). In the simplest case, c is the last hidden state $h_1 \ldots h_J$ of the encoder. Note that most modern encoders have two recurrent neural networks running in two directions (BiRNN) i.e., $h_j = [\overrightarrow{h_j}, \overleftarrow{h_j}]$. The decoder is also a recurrent neural network. It is fed several representations at each step i: the source representation c , the previous hidden state h_{i-1} and the target word previously predicted $E(e_{i-1})$. It generates a new hidden decoder state h_i and predicts a new target word e_i.

$$h_i = f(h_{i-1}, E(e_{i-1}), c_i) \tag{4.21}$$
$$o_i = \text{softmax}(W_1(W_2 h_{i-1} + W_3 E(e_{i-1}) + W_4 c_i)) \tag{4.22}$$

where f corresponds to the function computed by an RNN cell, that combining these inputs to generate the next hidden state. The output vector o_i conditioned on the decoder hidden state h_{i-1}, the embedding of the previous target word $E(e_{i-1})$ and c_i. In the simplest case, c_i is just the representation of the source sentence.

4.1.4.2 Attention mechanism

The motivation of this mechanism is to compute an association between the decoder state and each input word. Based on how relevant each particular input word is to produce the next output word, the model weighs the impact of its word representation. Bahdanau et al. [2015] add an alignment model (so-called "attention mechanism") to link generated output words to source words, which includes conditioning on the hidden state that produced the preceding target word. Luong et al. [2015] propose a "global" attention model, a variant of this mechanism and also a "local" attention model with hard constraints based on Gaussian distribution around a specific input word. In [Yang et al., 2017], they also use a recurrent neural network to model the attention mechanism, where a "dynamic memory" keeps track of the attention received by each source word, and demonstrate better translation results. Kim et al. [2017] introduce structural dependencies between source units. They show that structured attention networks outperform baseline attention models on a variety of tasks such as tree transduction, neural machine translation, question answering, and natural language inference.

The attention mechanism is achieved by computing a distinct context vector c_i (a position-dependent aggregated representation of the source) for each time step i of the decoding, before updating h_i and predicting a new target word e_i.

$$\alpha_{ij} = \frac{\exp(a(h_{i-1}, h_j))}{\sum_{k=1}^{J} \exp(a(h_{i-1}, h_k))} \tag{4.23}$$
$$c_i = \sum_j \alpha_{ij} h_j \tag{4.24}$$

Godard [2019] discusses two attention variants models `Attention (Update first)` and `Attention (Generate first)` based on the orders of two last phases in the decoder's RNN structure. In fact, Bahdanau et al. [2015] decompose the computations of the decoder in three phrases: Look, Generate and Update. The first phase *Look* is to find the context generating the current target word, the second phase *Generate* is to predict this target word, then followed by an update of the current hidden state *Update*. For `Attention (Update first)`, the *Update* phase is computed before *Generate* and the reversed order is used for `Attention (Generate first)`, implemented in [Sennrich et al., 2017]. Godard [2019] shows that updating first might conversely explain the one-position mismatch between attention and word alignment observed by Koehn and Knowles [2017a]. We also observe these two variants for the word alignment task.

Phase	Generate first	Update first
Look	$c_i \leftarrow h_{i-1}$	$c_i \leftarrow h_{i-1}, e_{i-1}$
Generate	$e_i \leftarrow h_{i-1}, e_{i-1}, c_i$	$e_i \leftarrow h_i, e_{i-1}, c_i$
Update	$h_i \leftarrow h_{i-1}, e_i, c_i$	$h_i \leftarrow h_{i-1}, e_{i-1}, c_i$

Table 4.1: Two variants of decoder's RNN structure

4.2 Neural alignment models

4.2.1 Non-probabilistic neural alignment models

Early work on the neural alignment model is in [Yang et al., 2013], which considers a feed-forward network to replace (and generalize) a conventional count-based translation model in an HMM model. They also give up the probabilistic interpretation which requires a softmax layer in the neural network to normalize overall words in a large size vocabulary. This helps them to avoid expensive computation for normalization. This line of work is continued by Tamura et al. [2014] who report an improvement by using recurrent neural networks. They assume that the recurrence helps to encode the entire history of previous alignments instead of only the last alignment. In short, their work aims to improve the alignment quality for a phrase-based translation system by using non-probabilistic scores.

[Legrand et al., 2016] tackle the problem differently by directly extracting the full word alignment matrix without using any underlying probabilistic model. They propose a matching score s_{ij} between a source word f_j and a target word e_i. This score is given by the dot-product.

$$s_{ij} = h_i^T h_j \tag{4.25}$$

where h_i and h_j are respectively the hidden states of e_i and f_j word. For unsupervised learning, they consider the aggregated matching scores over the source sentence between negative and positive sentence pairs. Note that a positive sentence pair includes two paired sentences whereas a negative sentence pair includes two unpaired sentences. This simple symmetrical approach has also proven useful for phrase-pair cleaning [Pham et al., 2018]. All these studies report AER scores and show improvements with respect to standard models, but lack a detailed analysis of the benefits of neural models in alignments.

Another line of research is alignment without parallel data. Sabet et al. [2020] propose a method of generating alignment links based on the matrix of embedding similarities. Note that they use mBert [Devlin et al., 2019] and the multilingual version of Fasttext[2] to generate multilingual embeddings from monolingual data.

4.2.2 Probabilistic neural alignment models

The work of Alkhouli et al. [2016], Wang et al. [2017] takes a different path, and explores ways to explicitly model alignments in NMT, revisiting with neural tools early word-based translation systems. In their approach, they study various neuralized models, some very similar to our word-based models (Section 4.3), of the standard alignment models, and also consider effective training strategies also exploiting weak supervision from count-based models.

[2]FastText is an open-source, free, lightweight library that allows users to learn text representations and text classifiers. https://fasttext.cc/

This line of research is continued by Deng et al. [2018], where attention vectors are processed as latent variables in NMT. The work of Rios et al. [2018] also exploits neural versions of conventional alignment (IBM-1/2) models, intending to improve word representations in low resource contexts; contrarily to most work focusing on NMT, some AER scores are reported, which are mostly in line with our neural baseline `IBM-1`. Note that we mainly follow this line of research by neuralizing distortion and translation models [Ngo-Ho and Yvon, 2019].

4.2.3 Word alignment from attention

A much more productive line of research tries to exploit the conceptual similarity between word alignments and attention [Koehn and Knowles, 2017b] to improve NMT. Mi et al. [2016], Liu et al. [2016], Chen et al. [2016], Alkhouli and Ney [2017] supervise the attention mechanism of recurrent models by putting the alignment cost to the NMT objective function. This cost is computed by calculating a distance between attentions and word alignments learned with alignment standard tools `Giza++` or `Fastalign`.

Cohn et al. [2016] modify the attention component to integrate some structural bias that has proved useful for alignments, such as a preference for monotonic alignments, for reduced fertility, etc. They also propose, following Liang et al. [2006], to enforce symmetrization constraints, an idea also explored in [Cheng et al., 2016, Li et al., 2018a]. Additional information about the to-be-aligned target word is used in [Peter et al., 2017, Li et al., 2018b] to improve attention models in terms of alignment accuracy. Note that different to alignment, the attention mechanism ignores the word to be aligned. Tu et al. [2016a] propose a coverage-based approach to reduce over-translation and under-translation problems. The same general methodology of using feature-based fertility is explored in [Luong et al., 2015, Feng et al., 2016, Yang et al., 2017] to introduce dependencies between adjacent alignment vectors. Garg et al. [2019], Zenkel et al. [2019] examine the effects of alignment on transformer models [Vaswani et al., 2017]. Note that they can extract alignment matrices from attention matrices by simply using a threshold. More work search for improving alignment and translation quality can be found in [Sankaran et al., 2016, Kuang et al., 2018, Ding et al., 2019b].

4.3 Variants of neural translation models

In Section 2.4, we mentioned that both `IBM-1` and `HMM` make the simplifying assumption that $p(f_j|f_1^{j-1}, a_1^j, e_1^I)$ simplifies to $p(f_j|e_{a_j})$. Analogous to these models, we propose two baseline neural variants `IBM-1+NN` and `HMM+NN`, where we implement the translation component with a neural network. As explained below, we then develop several additional versions, all relying on a simple and computationally efficient feed-forward architecture.

4.3.1 Context-free translation models

Our first neural model only modifies the translation model, keeping the distortion model unchanged with respect to the corresponding count-based version. Both the `IBM-1+NN` and `HMM+NN` use a simple feed-forward architecture which computes a distribution over possible source words f_j from an input target word e. In this architecture, a fixed size target vocabulary has to be specified to compute the softmax.

$$p_\theta(f_j|f_1^{j-1}, a_1^j, e_1^I) = p_\theta(f_j|e_{a_j}) \tag{4.26}$$

4.3.2 Contextual translation models

A first variant adds some context around the target word [Brunning et al., 2009, Collobert et al., 2011, Yang et al., 2013, Tamura et al., 2014, Abdel-Hamid et al., 2014, Alkhouli et al.,

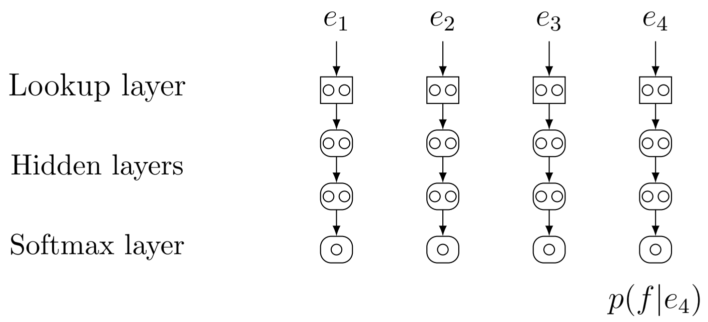


Figure 4.3: Structure of the context-free neural translation model `NN`

2016, Wang et al., 2017, 2018]. As the target words are fully observed, this modification has no impact on the computations needed to implement the model. We use a sliding window of size $(2*h+1)$ to represent word contexts and model $p(f_j|f_1^{j-1}, a_1^j, e_1^I)$ as $p(f_j|a_j, e_{a_j-h}^{a_j+h})$. For this variant, we compare two approaches to combine the embeddings of words in the context window:

- Concatenation (`NN+CtxCc`): We concatenate all word embeddings inside a window of size h and use a feed-forward layer for combination. We consider that the context of the null "word" is made of NULL tokens, similarly to [Yang et al., 2013].

- Convolution (`NN+CtxCnn`): We use a convolution filter of size $(2*h+1, 2*h+1)$ to combine context words. We use a simpler approach for the NULL model by performing a convolution over a window of NULL tokens.

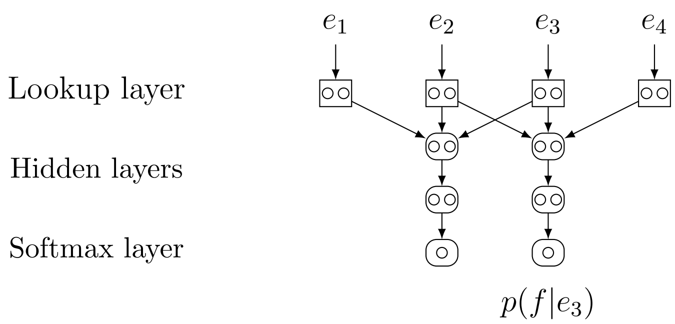


Figure 4.4: Structure of the contextual neural translation model

4.3.3 Character-based translation models

We consider ways to use character-based representations to improve or even replace word embeddings, so as to accommodate arbitrary vocabulary in source and target [Kang et al., 2011, Kim et al., 2015, Costa-jussà and Fonollosa, 2016, Labeau and Allauzen, 2017, Nicolai et al., 2018, Renduchintala et al., 2018, O'Neill and Bollegala, 2018, McCoy and Frank, 2018, Lan and Xu, 2018]. We apply a Bi-LSTM model to encode all characters in a target word e respectively in the forward $\overrightarrow{h_e}$ and backward $\overleftarrow{h_e}$ direction. We concatenate the resulting two hidden states $[\overrightarrow{h_e}, \overleftarrow{h_e}]$ to represent each target word. Again, three variants are considered:

- Pure character-based representations on the target side `NN+CharTgt`;

- Combined character-based and word-based representations on the target side `NN+CharWord`, where we simply concatenate both representations;

- Pure character-based representation on both sides `NN+CharBoth`: While the first two variants only amount to changing the target embeddings, this latter model is more challenging as we modify the source embeddings that are used in the output layer. While we keep a fixed size source vocabulary (i.e., 5000) in the softmax computation during training, we are in a position to compute the association of any source with any target word, known or unknown, during testing. In detail, we collect a full vocabulary V_b in a batch and also the most frequent words that have not been in v_b to obtain this fixed size source vocabulary.

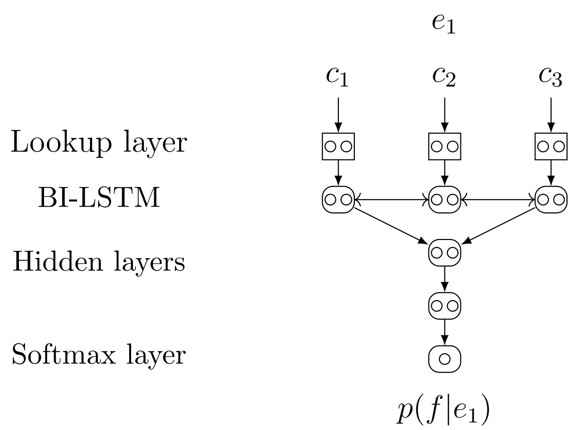


Figure 4.5: Structure of the character-based translation model: `NN+Char`

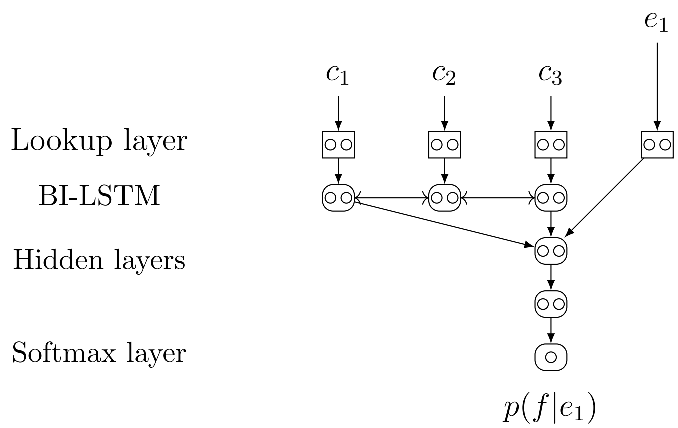


Figure 4.6: Structure of the character-based and word-based translation model: `NN+Char+Word`

4.4 Variants of neural distortion models

We mostly follow the assumptions of [Och and Ney, 2003] to design our distortion models. Only first-order dependencies are taken into account; furthermore, alignment positions only depend on the jump width and not on the absolute index position:

$$p(a_j|f_1^{j-1}, a_1^{j-1}, e_1^I) = p(\Delta_{a_j}) \tag{4.27}$$

where $\Delta_{a_j} = a_j - a_{j-1}$.

We restrict ourselves to jump values in the interval $[-K, +K]$ where K is a parameter of our model. For each sentence, the remaining probability mass corresponding to jumps greater than K or lower than $-K$ is uniformly divided among those valid offsets [Liang et al., 2006]. This means that we parameterize alignments using a multinomial distribution over $(2K+3)$ buckets. As a illustration, for $K = 1$, our jump distribution ranges over the five values:

$[\leq -1, -1, 0, 1, \geq 1]$. Note that we associate a specific NULL token to every target word, which allows us to faithfully model jumps from and to NULL tokens. The probability of transition to an empty word is governed by one single parameter p_0. Constraints for transitioning into and out of empty words follow the proposal of [Och and Ney, 2003]. For all variants of `IBM-1`, we thus use a uniform transition distribution $p(a_j|a_{j-1}) = \frac{1}{2I}$.

Variants of distortion models used in the `HMM` also rely on MLPs to compute the multinomial distribution in (4.27); they further combine character-based representations for word embeddings, as well as contextual word representations. Two settings are considered, where we only take the source, or the source and the target into account.

4.4.1 Character-based representation on the target side

Character-based representation on the target side `NNJumpTgt`: here the jump value only depends on characters of the target side [He, 2007]. We use the same character-based representations as above to represent words and also use a Bi-LSTM [Wang et al., 2018] to encode target word contexts. Therefore, the alignment probability becomes:

$$p(a_j|f_1^{j-1}, a_1^{j-1}, e_1^I) = p(\Delta_{a_j}|h_{a_{j-1}}) \tag{4.28}$$

where $h_{a_{j-1}}$ combines the forward and backward LSTM states computed for target word $e_{a_{j-1}}$, effectively encoding the full context around $e_{a_{j-1}}$.

4.4.2 Character-based representations on both sides

Character-based representations on both sides `NNJumpBoth`: we consider a more complex alignment model, which in addition takes into account the source side. Using the same representations as for the target side, we make the jump value also depend on the previously aligned source word. The source and target representations are concatenated before being passed through an MLP.

$$p(a_j|f_1^{j-1}, a_1^{j-1}, e_1^I) = p(\Delta_{a_j}|[h_{a_{j-1}}, h'_{j-1}]) \tag{4.29}$$

where h'_{j-1} is a context-dependent representation of the source word f_{j-1}.

Again, as source and target words are fully observed, these modifications have no impact on the computations used to compute the various quantities required for the estimation of our models. Finally note that in our implementation, the alignment and the translation models do not share any parameter.

4.5 Unsupervised Learning

In this framework, EM also applies [Berg-Kirkpatrick et al., 2010, Tran et al., 2016a]: during the (E) step, alignment posteriors are computed as usual using the Baum-Welch algorithm; in the (M) step, the main change is that the NN parameters have to be optimized numerically, e.g. via gradient descent.

Our training algorithm mostly follows [Tran et al., 2016a], where expectation-maximization (EM) is combined with back-propagation to train the neural network(s) models. For a number of training epochs, we repeat the following procedure:

1. For each batch:
   (a) Compute the posterior probability of each possible alignment link and the auxiliary function of the EM algorithm;
   (b) Improve the auxiliary function by performing one gradient update of the neural network parameters.

2. After a fixed number of batches: (a) For discrete distortion models, collect and store the entire translation model and jump width distribution for all sentences in the corpus; update the jump distribution. (b) For neural distortion models, collect and store the entire translation and distortion models for computing posterior probabilities.

The initial parameter values are either random (for `IBM-1`) or are initialized with the parameter values of the corresponding `IBM-1` models (for the `HMM` models). Note that we could perform more than one gradient update for each batch as in [Tran et al., 2016b]. In our initial experiments, we found that this approach did not significantly improve the AER score after 10 iterations. Moreover updating gradients is computationally expensive. Therefore, we decided to stick with one gradient update for each batch.

4.6 Experiments

We compute the performance of our models after 10 EM iterations. We collect and store the parameters of the translation and distortion model after 50 batches. Note that we shuffle all sentences after each iteration and create batches by sorting a few consecutive sentences [Morishita et al., 2017]. Our optimizer is Adam [Kingma and Ba, 2014] with an initial learning rate of 0.001. The batch size is set to 100 sentences. We use all sentences of length lower than 50 and a 50K word vocabulary for both the source and target languages. Note that for English-Vietnamese, we use a full vocabulary in their training corpus, i.e., 42 544 and 19 853 words for respectively English and Vietnamese. We highlight that different to the baselines, we have separate training set and test set.

Our neural translation models are based on a simple architecture composed of a word embedding layer (64 units), feed-forward layers (each comprising 64 units) with activation function htanh [Yang et al., 2013], followed by a drop-out layer and a softmax layer. The contextual models use a context window of size $h = 1$, based on the experiments reported in Tamura et al. [2014]. For the character-based models, the Bi-LSTM model also contains 64 units in the embedding layers and in the hidden layers.

In the discrete alignment model, we consider jump values in the interval $[-5, +5]$ [Liang et al., 2006]. Note that in [Yang et al., 2013], their lexicalized distortion does not produce better alignment than the simple discrete alignment model on small scale data. In the neural alignment models, the interval is $[-80, +80]$. For the convolutional models, we apply one small filter of size (3,3) to combine context word embeddings.

In our experiments, we mainly use Python version 3.6, Numpy version 1.2, Tensorflow version 1.0.1 and Pytorch version 0.4.1. The implementation is available from `https://github.com/ngohoanhkhoa/Generative_Probabilistic_Alignment_Models`.

Datasets and 50K word vocabulary Table 4.2 shows the basic statistics for unknown words in the case where the sentence length is lower than 50 and the vocabulary size is 50K. In other words, we report the number of words which are not the top 50K frequent and the corresponding number of unknown types in parentheses. For example, there are 79 different unknown words (for the vocabulary size 50K) that occurs 176 times in the English test corpus for English-French language pair. The out-of-vocabulary word is denoted UNK. This selection of a 50K word vocabulary can be found in many works for NMT such as Luong et al. [2015], Jean et al. [2015], Luong and Manning [2016], See et al. [2016], etc. Note that the number of unknown words in the test set increases remarkably under the 50K word vocabulary, specially in the case of English-Czech. We consider the baselines that use a complete vocabulary for training (much larger than the vocabulary size of the neural models) for all analysis. In the unknown word analysis, we also consider the case where the baselines use a 50K word vocabulary. We expect that the alignment for unknown words is improved by considering context around this UNK token (`NN+Ctx`). Moreover, using character-based models (`NN+Char`), specially the variant

NN+CharBoth helps to get rid of the unknown word problem. Another approach to deal with this vocabulary size is to use subwords (BPE in Section 2.3.2) that we will discuss in Chapter 6.

Test corpus	# unk words in test set	
	En	Fr
English-French	176 (79)	104 (79)
English-German	26 (26)	187(180)
English-Romanian	43 (37)	166 (150)
English-Czech	1 911 (1 073)	5 170 (3 851)
English-Japanese	874 (655)	495 (379)
English-Vietnamese	13 927 (3 866)	12 335 (2 362)
English-Romanian Dev	14 (13)	133 (131)

Table 4.2: Basic statistics for unknown words in the test corpora under the condition of sentence length (< 50 words) and of vocabulary size 50K.

4.6.1 Hyper-parameter settings

We search the appropriate configuration for all of our models, based on the results given by the English-Romanian development set (a small scale data) after 50 EM iterations.

Word embedding size and hidden layer size We explore the number of word embedding units and feed-forward units by observing the AER of IBM-1+NN. This model includes a word embedding layer, a feed-forward layer, followed by a drop-out layer and finally a softmax layer. As can be seen in Figure 4.7, we find that using a larger number of embedding cells (> 64 units) did slightly improve the AER score (< -0.005 AER) after 10 iterations. As for the other meta-parameters, we decided to stick with these baseline values: we assume that the relative differences between models observed in our setting would carry-over, albeit with slightly different values, for larger models.

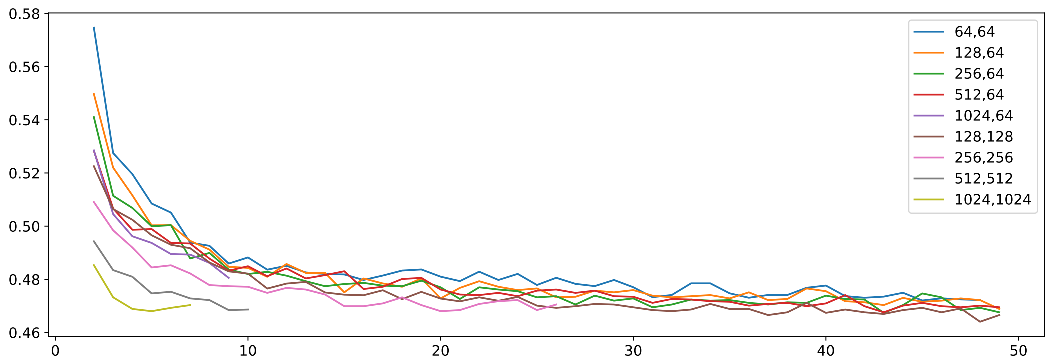


Figure 4.7: Model configurations: AER of IBM-1+NN with the different configurations. Each configuration is a pair of unit numbers (the former is the word embedding units, the latter is the feed-forward units). The x-axis shows the number of iterations. The y-axis represents the AER.

Number of layers The AER of the models with the different numbers of layers (each comprising 64 units) are given in Figure 4.8. As can be seen, our models do not benefit from a larger number of layers. Therefore, our vanilla model includes two feed-forward layers.

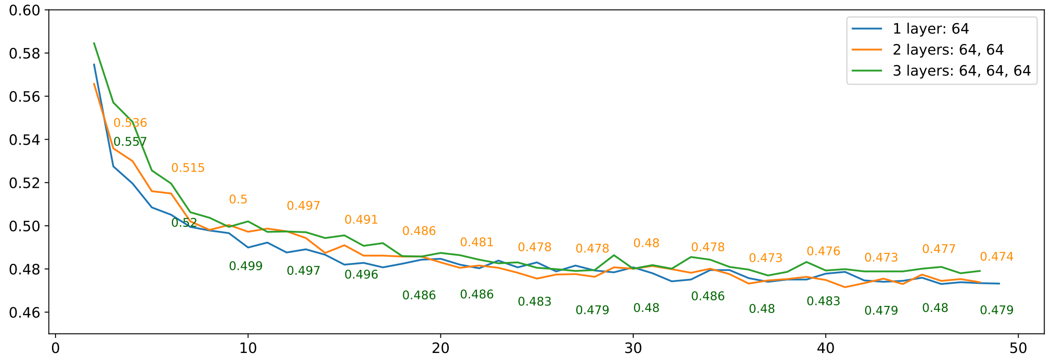


Figure 4.8: Model configurations: AER of `IBM-1+NN` with different numbers of layers. The x-axis shows the number of iterations. The y-axis represents the AER. We compare the three different configurations including 1, 2 and 3 hidden layers.

Vocabulary size of 50K, is it a problem for our models ? We observe the performance of the full vocabulary in Figure 4.9. The differences between 50K words and all words in vocabulary are not remarkable, which means that this vocabulary size, however small, is an appropriate vocabulary size.

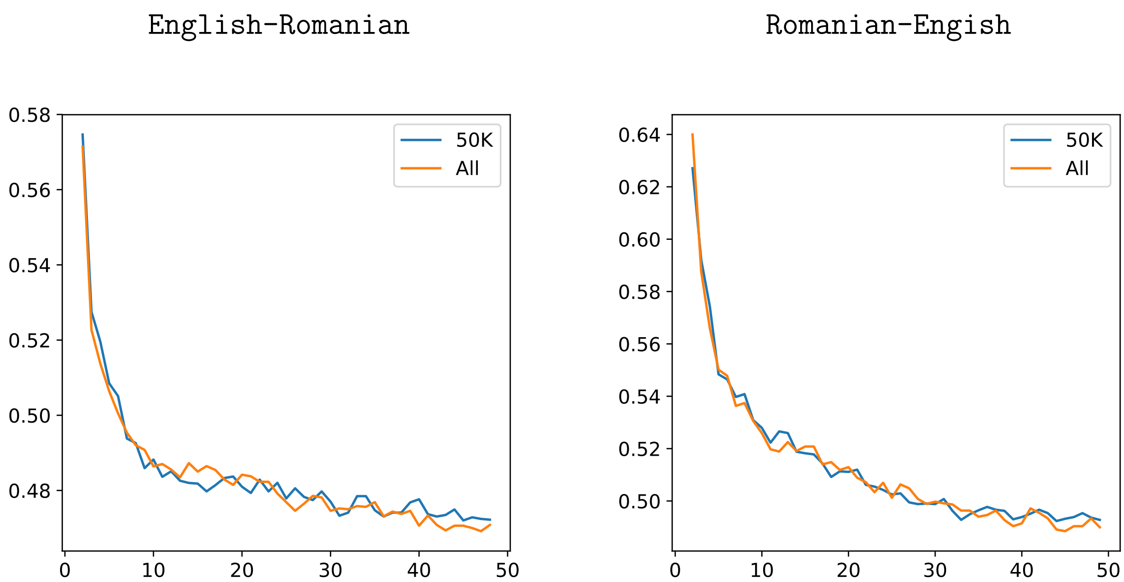


Figure 4.9: Model configurations: AER of `IBM-1+NN` with 50K words and all words in vocabulary. The x-axis shows the number of iterations. The axis y represents the AER.

4.6.2 Experiments with attention-based models

In order to observe the behavior of alignment links generated by attention models (Section 4.2.3), we use the implementations of the two attention-based models [Godard, 2019]: Attention (Update first) `U` and Attention (Generate first) `G` (Section 4.1.4.2). We extract alignment links from a machine translation task: the results of these models are matrices of attention showing the probability of source sentence's words for each target word. In order to generate an alignment matrix, we apply two simple approaches:

- Argmax: We select only one source word having the highest probability. This model is still an asymmetrical model. This yields the two models `GA` and `UA`.

- Threshold: We select all source words having their probabilities higher than a threshold and fine-tune this threshold to achieve the best AER score. The threshold of 0.2 is used in our experiments. This yields the two models `GT` and `UT`. Note that using a threshold enables to generate many-to-many links.

An example of these two approaches is displayed in Figure 4.10.

Argmax

	f_1	f_2	f_3	f_4	f_5
e_1	0.5	0.1	0.1	0.1	0.2
e_2	0.1	0.3	0.4	0.1	0.1
e_3	0.1	0.3	0.1	0.4	0.1
e_4	0.2	0.1	0.2	0.2	0.3

Threshold equal to 0.2

	f_1	f_2	f_3	f_4	f_5
e_1	0.5	0.1	0.1	0.1	0.2
e_2	0.1	0.3	0.4	0.1	0.1
e_3	0.1	0.3	0.1	0.4	0.1
e_4	0.2	0.1	0.2	0.2	0.3

Figure 4.10: Example of the two simple approaches (Argmax and Threshold) that help to generate an alignment matrix from an attention matrix. Cells in dark are retained in the final alignment.

4.7 Evaluation

In this section, we perform a detailed analysis of using the quantitative metrics presented in Chapter 3, focusing mostly on the differences between discrete and neural versions of the `HMM` and `IBM` models. Our goal in this section is to better understand the improvements brought by the neural models, but also to highlight the problems that remain difficult for alignment models. Complete results are in [Ngo Ho, 2021, Appendix B]. We also report the performance of attention-based models (Section 4.2.3) for the alignment task and their complete results are shown in [Ngo Ho, 2021, Appendix C].

4.7.1 AER, F-score, precision and recall

We reports the scores of our neural models (`IBM-1+NN`, `HMM+NN` and their variants) and also our four baselines (`IBM-1`, `HMM`, `IBM-4` implemented in `Giza++`, `Fastalign`) [Ngo Ho, 2021, Appendix B.1]. We also compare our best results with other published numbers for English vs French, German, Romanian and Japanese in [Ngo Ho, 2021, Appendix F]. Note that for English-French and English-Romanian, the training corpora used in related works are different from ours (see details in Section 3.1.1), we hence report the results of our models for these corpora.

As can be seen in Table 4.3, a first general observation is that almost all neural network models outperform their discrete counterpart, with our best `HMM` models even outperforming `IBM-4` for almost all language pairs. The improvements are overall lesser for German: on the one hand, the issues with unknown words are not as bad as for Czech, owing to a larger training set (see Table 3.3); on the other hand all our NN architectures fail to improve the modeling of alignments of German compounds which typically yield many-to-one alignment links that are poorly predicted (see Figure 4.11); word order differences with English are another area where our models do not help much. We deepen our analysis of German in Section 4.7.9.

Most of the improvement is already achieved by the vanilla NN model (Table 4.4), which improves over the baseline for all languages, sometimes for a very large margin, e.g. -8/9 AER

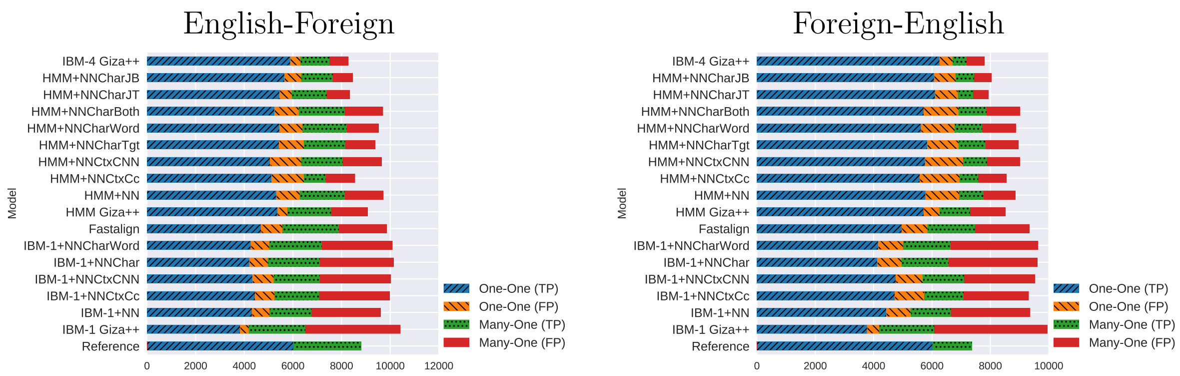


Figure 4.11: Results of our neural models: Alignment types for English-German

Corpus	IBM-1 Giza++	IBM-1+NNs			Fastalign	HMM Giza++	IBM-4 Giza++	HMM+NNs		
		#	Best	AER				#	Best	AER
English-French	40.1	5/5	NNCharWord	27.03	15.19	11.99	10.00	8/8, 8/8, 3/8	NNCharJT	8.41
French-English	33.9	5/5	NN vanilla	27.21	16.23	11.97	9.64	8/8, 7/8, 3/8	NNCharJT	7.70
English-German	39.03	5/5	NNCharWord	35.31	28.98	23.92	21.46	6/8, 2/8, 0/8	NNCharJB	23.69
German-English	42.66	5/5	NNCtxCNN	36.02	31.28	26.33	23.31	8/8, 2/8, 0/8	NNCharJB	24.90
English-Romanian	56.02	5/5	NNCtxCNN	46.15	33.36	33.36	31.04	7/8, 7/8, 7/8	NNCharWord	25.51
Romanian-English	53.52	5/5	NNCtxCNN	43.93	32.91	36.38	32.30	6/8, 7/8, 6/8	NNCharTgt	28.01
English-Czech	45.09	4/5	NNCharWord	40.28	25.75	27.86	20.92	8/8, 8/8, 5/8	NNCharJT	15.94
Czech-English	48.47	5/5	NN vanilla	40.47	25.30	30.38	26.50	8/8, 8/8, 7/8	NNCharWord	22.80
English-Japanese	63.12	5/5	NNChar	57.96	50.67	57.01	52.52	8/8, 8/8, 8/8	NNCharJT	39.69
Japanese-English	61.55	5/5	NNCharWord	53.54	49.37	54.41	49.23	8/8, 8/8, 8/8	NNCharJB	37.71
English-Vietnamese	69.43	5/5	NNCharWord	53.2	48.89	57.86	51.91	5/8, 8/8, 8/8	NNCharJB	43.28
Vietnamese-English	46.45	5/5	NNCharWord	35.45	32.82	37.57	33.19	8/8, 8/8, 8/8	NNCharJB	27.59

Table 4.3: Best AER of our NN models compared with the corresponding baselines. We report the number of NN models that outperform their counterpart (#), the name of the NN model that obtains the best AER (Best) among the NN models and its score (AER). In the case of HMM, there are three numbers representing the number of `HMM+NN` models respectively outperforming `Fastalign`, `HMM Giza++` and `IBM-4 Giza++`.

for the neural IBM-1 for the pair Romanian-English in both directions. The corresponding gains of the basic neural HMM model are large for English vs Czech (-5 AER), Japanese (-7 AER) and Vietnamese (-6 AER).

Corpus	IBM-1 Giza++	IBM-1+NN	Fastalign	HMM Giza++	HMM+NN
English-French	40.1	27.96	15.19	11.99	11.84
French-English	33.9	27.21	16.23	11.97	11.15
English-German	39.03	37.64	28.98	23.92	26.78
German-English	42.66	39.22	31.28	26.33	29.44
English-Romanian	56.02	46.4	33.36	33.36	30.69
Romanian-English	53.52	44.9	32.91	36.38	40.12
English-Czech	45.09	42.29	25.75	27.86	23.5
Czech-English	48.47	40.97	25.3	30.38	24.06
English-Japanese	63.12	62.64	50.67	57.01	49.68
Japanese-English	61.55	56.9	49.37	54.41	47.09
English-Vietnamese	69.43	58.87	48.89	57.86	49.27
Vietnamese-English	46.45	42.25	32.82	37.57	31.45

Table 4.4: AER of our NN vanilla models (Section 4.3.1) compared with our baselines.

Regarding contextual variants, a first observation is that the difference between concatenation and convolutions is limited, typically in the order of 1 AER point; the latter approach seems to be on average the best choice. A comparison with the neural `IBM-1` baselines reveals that the contextual version is not always better than the default. The largest gains are observed in small data conditions (Romanian/Czech/Japanese/Vietnamese-English) when English is on the target side. The scores of `+NNCtx` for English-Romanian are in Table 4.5. In this case, the context helps to disambiguate alignment links for English words by improving the translation distribution $p(f_j|a_j, e_{a_j-h}^{a_j+h})$. For instance, we find that the context vastly improves the precision (from 49.92% to about 62.73%) as well as the recall (from 43.5% to about 51.64%) in the direction Romanian-English; in the other direction, the change is insignificant. Similar behavior is found for the variants of `HMM+NNCtx`. Compared with `HMM+NN`, the gain (of about -1/2 AER points) is often made by `HMM+NNCtxCNN` for some directions e.g., Czech-English; English-Japanese and English-Vietnamese (in both directions). In the direction Romanian-English, we notice a large improvement of about -9 AER points just because `HMM+NN` does not work well.

Baselines	English-Foreign					Foreign-English				
	AER	F1	PRE	REC	ACC	AER	F1	PRE	REC	ACC
`IBM-1 Giza++`	56.02	43.99	58.8	35.14	96.66	53.52	46.49	**49.92**	**43.5**	96.26
`IBM-1+NN`	46.4	53.62	57.71	50.07	96.77	44.9	55.11	60.08	50.9	96.91
`IBM-1+NNCtxCc`	49.93	50.09	54.28	46.49	96.54	43.95	56.07	61.32	**51.64**	96.98
`IBM-1+NNCtxCNN`	46.15	53.87	60.08	48.81	96.88	43.93	56.09	**62.73**	50.72	97.04
`Fastalign`	33.36	66.65	72.77	61.49	97.7	32.91	67.1	73.7	61.59	97.75
`HMM Giza++`	33.36	66.65	75.28	59.8	97.77	36.38	63.64	72.9	56.46	97.59
`HMM+NN`	30.69	69.33	76.93	63.09	97.92	**40.12**	59.89	63.85	56.4	97.18
`HMM+NNCtxCc`	33.84	66.18	75.21	59.08	97.75	34.83	65.19	73.24	58.73	97.66
`HMM+NNCtxCNN`	30.87	69.15	80.01	60.89	97.97	**31.03**	68.99	77.58	62.11	97.92
`IBM-4 Giza++`	31.04	68.98	79.28	61.04	97.95	32.3	67.72	80.97	58.2	97.93

Table 4.5: Alignment error rate (AER), accuracy (ACC), F-score (F1), precision (PRE) and recall (REC) for English-Romanian. This is for contextual models.

Models using character-based in the target (with or without word information) also yield significant and consistent gains, especially also in small data conditions. Comparing the two conditions, we see that combining word and character information is not always the best approach, as the pure character-based approach is sometimes even better. Our claim is that the pure character-based approach should be preferred given a sufficiently large dataset (as in the English-French condition Table 4.7); when this is not the case, word information, which is easier to train, can also prove helpful. We also notice that the model `+NNCharWord` produces a slightly larger recall, leading a better F-score in most cases.

With respect to the neural baseline, the gains are maximal when the morphologically rich language (e.g. Czech) is on the target side: in this situation, character-based representations help to improve the translation model for the rare words, which in the other versions all correspond to the same UNK symbol[3]. An illustration for this is displayed in Table 4.6 for English-Czech. The gain (about -7 AER) in the direction English-Czech is larger than in the opposite direction. This is because there are more unknown words in Czech (5 170 words) than in English (1 073 words), and eliminating UNK symbol clearly helps. The effect is less clear for French because the number of unknown English and French words is small (Section 4.2).

The use of character models in the source side did not enable us to improve these results. Our claim is that using a vocabulary of 5000 words in the softmax computation is not good enough to overcome the unknown source word problem. A larger vocabulary can helps but it requires more expensive computational cost of training. In fact, after each parameter update

[3]Remember that the neural models, contrarily to the discrete models, use a limited vocabulary of 50K words.

step, all source word representations must be recomputed. Therefore, we do not explore more this technique and we consider another approach (i.e., BPE) to eliminate unknown words.

Models	English-Foreign					Foreign-English				
	AER	F1	PRE	REC	ACC	AER	F1	PRE	REC	ACC
IBM-1+NN	42.29	48.64	54.32	44.04	96.22	40.97	49.08	56.81	43.2	96.36
IBM-1+NNChar	**40.85**	50.35	54.2	47.01	96.24	42.35	47.99	55.68	42.17	96.29
IBM-1+NNCharWord	**40.28**	50.82	55.18	47.11	96.3	46.2	44.94	51.87	39.65	96.06
HMM+NN	23.5	65.45	74.39	58.43	97.5	24.06	64.03	75.48	55.6	97.46
HMM+NNCharTgt	**16.74**	69.36	83.82	59.15	97.88	24.61	63.94	73.42	56.63	97.41
HMM+NNCharWord	**16.04**	70.34	83.18	60.93	97.91	22.8	64.94	77.4	55.94	97.55
HMM+NNCharBoth	**17.38**	69.09	81.89	59.76	97.83	28.26	61.04	70.15	54.02	97.2
IBM-4 Giza++	20.92	65.7	79.48	56	97.63	26.5	59.81	75.58	49.48	97.3

Table 4.6: Alignment error rate (AER), accuracy (ACC), F-score (F1), precision (PRE) and recall (REC) for English-Czech. This is for character-based models.

Regarding distortion models, we see a gain in using a neuralized version of the jump model in the cases where character-based models are already helping. This gain can be well observed for German, Japanese and Vietnamese [Ngo Ho, 2021, Appendix B.1]. For example, in Table 4.8 (English-German), neural distortion models improve both AER (-3 points) and F-score (+3 points). Moreover, we see also nice gains in precision (+9 points). For English-French (Table 4.7) where there is a large number of possible links, the loss in recall yields better AERs but worse F-scores. An explanation is that for these languages our neural distortion models help to correctly predict unaligned words. For Czech and Romanian, we do not find similar improvements in our setting. We discuss this situation in Section 4.7.3 and also Section 4.7.4.

Models	English-Foreign					Foreign-English				
	AER	F1	PRE	REC	ACC	AER	F1	PRE	REC	ACC
IBM-1 Giza++	40.1	26.7	71.55	16.41	89.01	33.9	36.49	59.24	26.37	88.81
IBM-1+NN	27.96	36.42	69.66	24.65	89.5	**27.21**	38.08	68.12	26.43	89.52
IBM-1+NNChar	28.76	37.5	67.13	26.01	89.42	31.4	37.21	62.64	26.47	89.11
IBM-1+NNCharWord	**27.03**	38.69	68.96	26.89	89.61	28.33	38.97	66.1	27.63	89.45
Fastalign	15.19	44.98	82.5	30.92	90.78	16.23	46.32	80.08	32.58	90.79
HMM Giza++	11.99	45.18	86.12	30.62	90.94	11.97	45.98	85.2	31.49	90.98
HMM+NN	11.84	45.57	86.68	30.91	91	11.15	46.86	86.17	32.18	91.1
HMM+NNCharTgt	9.17	47.22	89.53	32.07	91.26	9.56	47.87	88.15	32.86	91.27
HMM+NNCharWord	10.45	47.33	87.94	32.38	91.21	10.27	48.56	86.92	33.69	91.3
HMM+NNCharBoth	10.9	46.74	87.41	31.9	91.13	11.17	47.5	86.1	32.8	91.16
HMM+NNCharJT	**8.41**	44.71	91.58	29.57	91.08	**7.70**	44.45	92.82	29.22	91.09
HMM+NNCharJB	8.47	44.38	91.8	29.26	91.06	7.74	46.26	91.42	30.96	91.23
IBM-4 Giza++	10	44.43	90.61	29.43	91.02	9.64	45.43	89.58	30.43	91.08

Table 4.7: Alignment error rate (AER), accuracy (ACC), F-score (F1), precision (PRE) and recall (REC) for English-French. This is for neuralized distortion models.

We compare our neural models with the attention-based models. Complete results are found in [Ngo Ho, 2021, Appendix C.1]. We report the results for English-Romanian in Table 4.9. The attention-based model (Generate first) `G` shows the slight improvements compared with `IBM-1`, whereas `U` (Update first) is much worse. One reason could be the context vector of the current target word e_i is computed with the ground-truth target word at e_{i-2} (`U`) instead of e_{i-1} (`G`). This explains the one-position mismatch between attention and word alignment shown in

Models	English-Foreign					Foreign-English				
	AER	F1	PRE	REC	ACC	AER	F1	PRE	REC	ACC
Fastalign	28.98	68.75	71.11	66.54	97.35	31.28	66.47	70.73	62.69	97.23
HMM Giza++	**23.92**	73.3	79.23	68.2	97.82	**26.33**	71.04	79.47	64.23	97.7
HMM+NN	26.78	70.95	73.94	68.2	97.55	29.44	68.21	74.69	62.76	97.44
HMM+NNCharTgt	26.04	71.57	75.99	67.64	97.64	28.11	69.48	75.59	64.29	97.52
HMM+NNCharBoth	27.14	70.6	73.65	67.79	97.52	29.31	68.34	74.11	63.41	97.42
HMM+NNCharJT	23.79	73.15	82.8	65.52	97.89	**25.21**	71.85	83.64	62.98	97.84
HMM+NNCharJB	**23.69**	73.38	82.38	66.15	97.9	**24.9**	72.16	83.36	63.61	97.85
IBM-4 Giza++	21.46	75.48	85.79	67.39	98.08	23.31	73.63	86.56	64.06	97.99

Table 4.8: Alignment error rate (AER), accuracy (ACC), F-score (F1), precision (PRE) and recall (REC) for English-German. This is for character-based models.

Koehn and Knowles [2017a]. We also see that GT using a threshold has higher recall than its counterparts.

Models	English-Foreign					Foreign-English				
	AER	F1	PRE	REC	ACC	AER	F1	PRE	REC	ACC
IBM-1 Giza++	56.02	43.99	58.8	35.14	96.66	53.52	46.49	49.92	43.5	96.26
IBM-1+NN	46.4	53.62	57.71	50.07	96.77	44.9	55.11	60.08	50.9	96.91
IBM-1+NNCtxCc	49.93	50.09	54.28	46.49	96.54	43.95	56.07	61.32	51.64	96.98
IBM-1+NNCtxCNN	46.15	53.87	60.08	48.81	96.88	43.93	56.09	62.73	50.72	97.04
IBM-1+NNChar	50.16	49.85	54.28	46.09	96.54	48.28	51.73	56.08	48.01	96.66
IBM-1+NNCharWord	46.54	53.47	56.91	50.43	96.73	43.94	56.08	60.71	52.1	96.96
GA	**50.71**	49.31	51.71	47.11	96.39	46.98	53.03	56.39	50.05	96.69
GT	**49.1**	50.91	50.82	**51**	96.33	45.1	54.92	58.24	**51.95**	96.82
UA	63.36	36.65	38.44	35.02	95.48	63.35	36.66	38.98	34.6	95.54
UT	59.43	40.58	35.97	46.54	94.91	59.93	40.08	34.81	47.24	94.73

Table 4.9: Alignment error rate (AER), accuracy (ACC), F-score (F1), precision (PRE) and recall (REC) of English-Romanian. This is for attention-based models.

4.7.2 Do neural networks improve performance for long sentences?

The AER scores of our neural models for varying sentence length and sentence length difference are respectively in [Ngo Ho, 2021, Appendix B.10] and [Ngo Ho, 2021, Appendix B.11]. We see a clear benefit of neural networks for sentences having more than 80 words. Note that our training set uses only sentences of length lower than 50. In the case of English-Czech (Figure 4.12), for long sentences, the AER can be as high as 70 AER for IBM-4 Giza++ whereas the highest error rate of HMM+NN is about 50 AER. This improvement is found also in Romanian, Japanese and Vietnamese for both directions. For French and German, their testing sets do not include such long sentences but we also observe similar gains.

Figure 4.12: The direction English-Czech: AER score as a function of sentence length. The x-axis shows the sentence length. The y-axis represents the AER. The annotation displays the number of sentences.

4.7.3 How do neural models process unaligned words?

We study the accuracy of alignment models [Ngo Ho, 2021, Appendix B.2] and patterns of unaligned words [Ngo Ho, 2021, Appendix B.3]. For example, in the case of English-Czech, we discuss our 17 models based on the correct/incorrect alignment links (Figure 4.13) and the correct/incorrect unaligned words (Figure 4.14).

As can be observed in Figure 4.14, the models in the `IBM-1` family generate very few unaligned source words (null links), and concentrate all their efforts in generating correct (or wrong) links between actual words (Figure 4.13) as already noted by Moore [2004]. Variants of the `HMM` model display a different pattern:

- They make fewer predictions (and fewer errors) for non-null links.
- They tend to predict a large number of null links, with only a small portion of them being actually correct i.e., a large number of the incorrect unaligned words.

The latter effect is less clear in the case of Japanese and Vietnamese which contain a large number of unaligned source words in their alignment references (see statistics in Section 3.3 and results in [Ngo Ho, 2021, Appendix B.3]).

About half of the remaining errors of our best models concern null links, in this case the prediction of a link for a word that should have stayed unaligned. Similar trends were observed for the other language pairs/directions. Null links are often due to deep to syntactic divergences between languages and are quite hard to predict based on the sole source word. This is mostly a modeling issue, for which the transition from discrete to neural models is of little help in precision for null links. Figure 4.14 displays the number of unaligned/aligned words for the variants of `HMM` and for English-Czech and English-Vietnamese. Our two models using neural distortion models, `HMM+NNCharJT` and `HMM+NNCharJB`, predict more correctly unaligned words for English-Vietnamese (the number of correctly unaligned words is larger than the number of incorrectly unaligned words). This kind of improvement can be found in the language pairs containing a large number of null links. However, over-generating null links can be harmful e.g., `HMM+NNCharJT` for the direction English-Czech.

To sum up, we see the clear benefits of using neural translation models: both for the `IBM-1` variants and the `HMM` variants yield a clear reduction of errors (Incorrect alignment (FP) links and incorrect null links (FN) as can be seen in Figure 4.13).

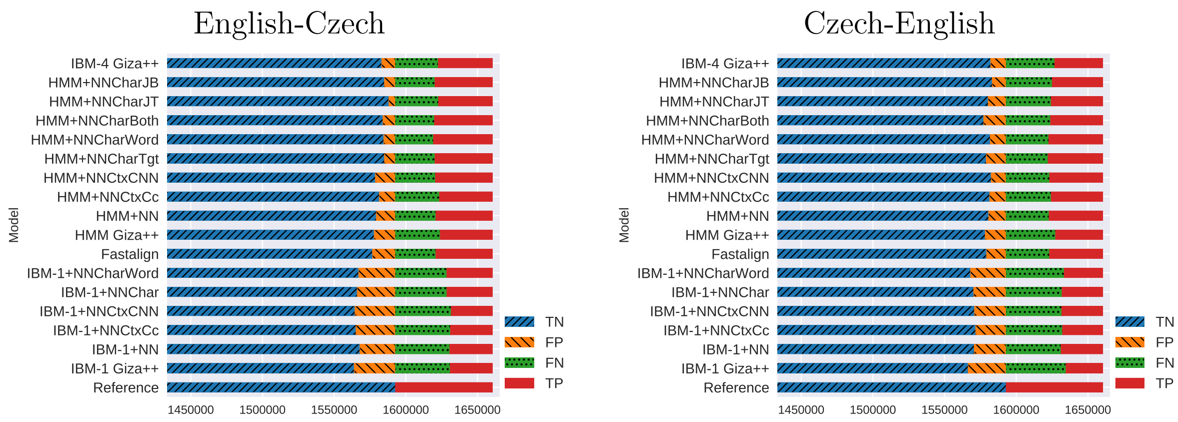


Figure 4.13: Results of alignment links for English-Czech in both directions: We see that `IBM-1` family has more FP/FN and less TN than the variants of the `HMM`. In the language pair English-Vietnamese, `HMM+NNCharJT` and `HMM+NNCharJB` obtain some more correctly unaligned words than `HMM+NNCharWord`.

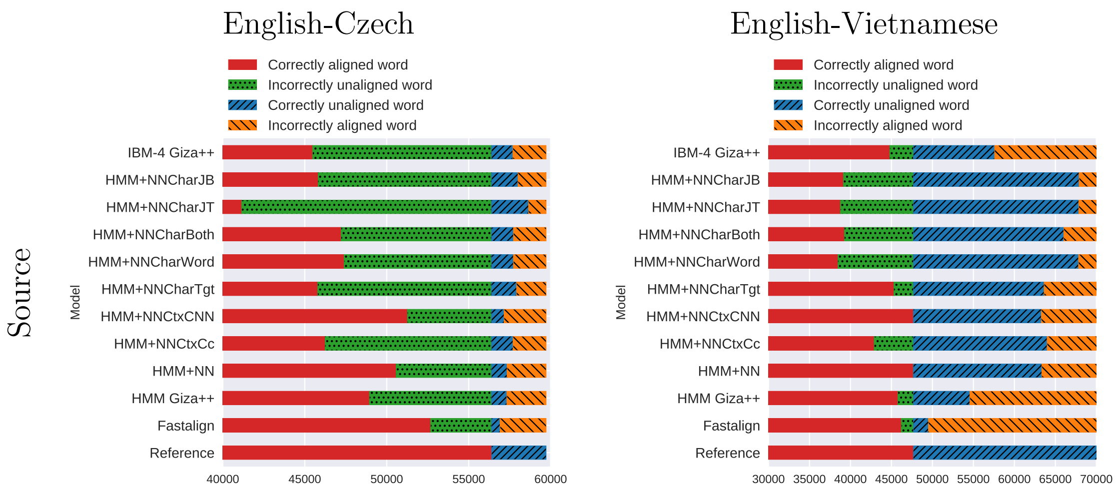


Figure 4.14: Results of unaligned source words for the variants of `HMM` in the two cases: the direction English-Czech and the direction English-Vietnamese.

4.7.4 Is word distortion improved by neural networks ?

In our implementations of neural alignment models, we first vary the translation model, leaving the distortion model unchanged, which allows us to single out the effect of using a stronger translation model. We then neuralize distortion models to seek more important improvements. Complete results are in [Ngo Ho, 2021, Appendix B.5].

In general, we see that `IBM-1` over-generate links in three areas: short jumps (with jumps equal to 0 or 1), and long jumps, greater than 5 positions in either direction. Its neuronal counterpart amplifies this tendency to over-predict long jumps (Figure 4.15). We observe the behaviors of the neuronal `HMM` models which reflect two patterns of word distortion with clear differences between European languages and Asian languages (Section 3.5):

- European languages: The neuronal `HMM` models tend to generate too many short jumps equal to 1, as well as too many null alignments while `Fastalign` has a much more even distribution of jumps. An illustration for German is in the top graph of Figure 4.15.
- Asian languages: We see the same short jump effect where too many short jumps are equal to 0. Example of Japanese is in the bottom graph of Figure 4.15.

Similar trends are found in other directions/language pairs. To sum up, the above-mentioned observations suggest that much remains to be done in terms of better modeling the distortion, our best models having a tendency to concentrate the link distribution around short jumps, a likely sign of a too confident translation model.

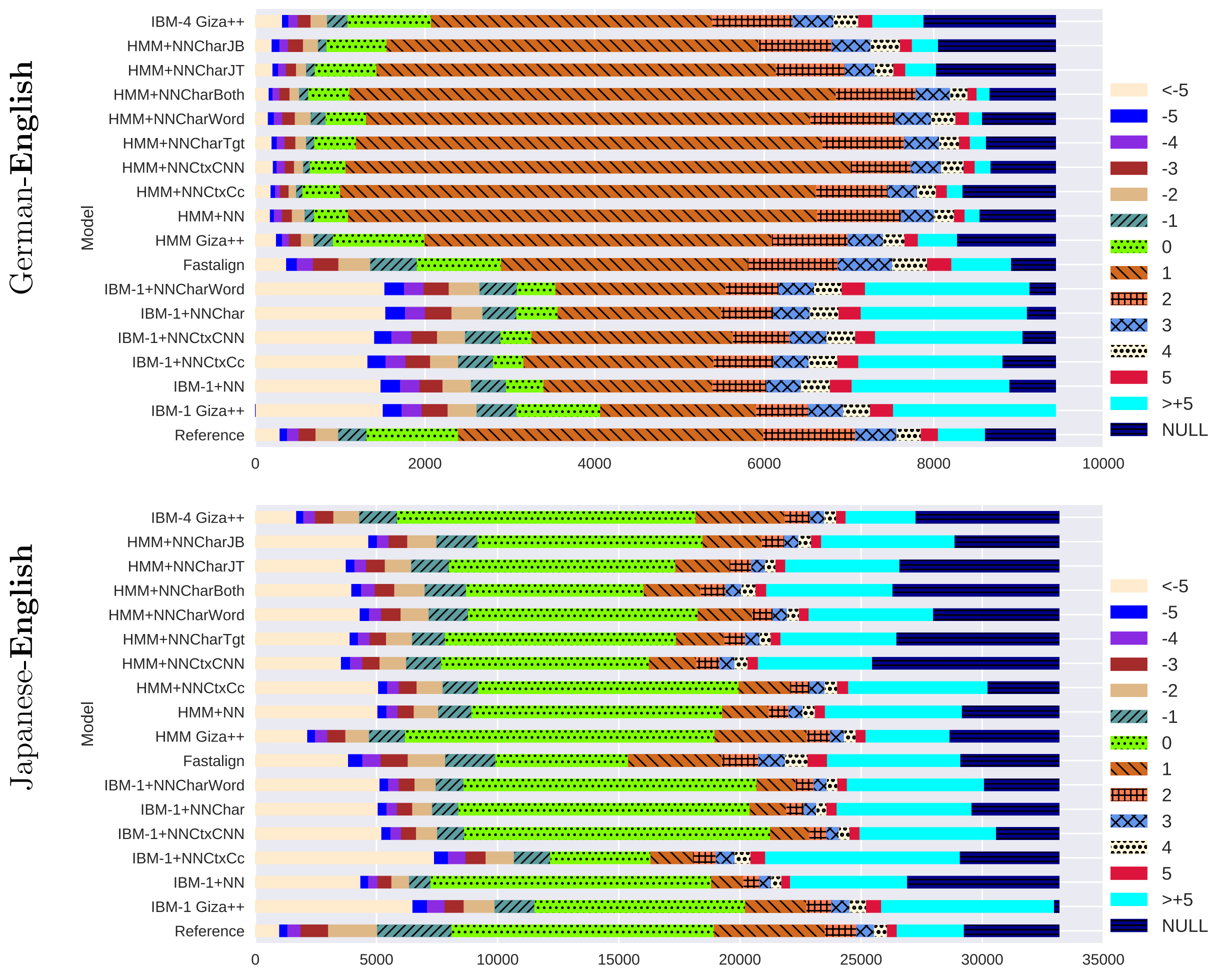


Figure 4.15: Jump widths for English words for the direction German-English and for the direction Japanese-English

Our neural distortion models however slightly improve the performance in two ways. We can see it in Figure 4.16 displaying the number of correct/incorrect jumps for English-German.

- They generate more correct jumps of length 1, which clearly helps to improve the precision and also the F-score.
- The number of null alignments increases. Most null links are incorrect, which harms the recall. However, there is also a rise of correct null link numbers, helping to gain some more points of AER.

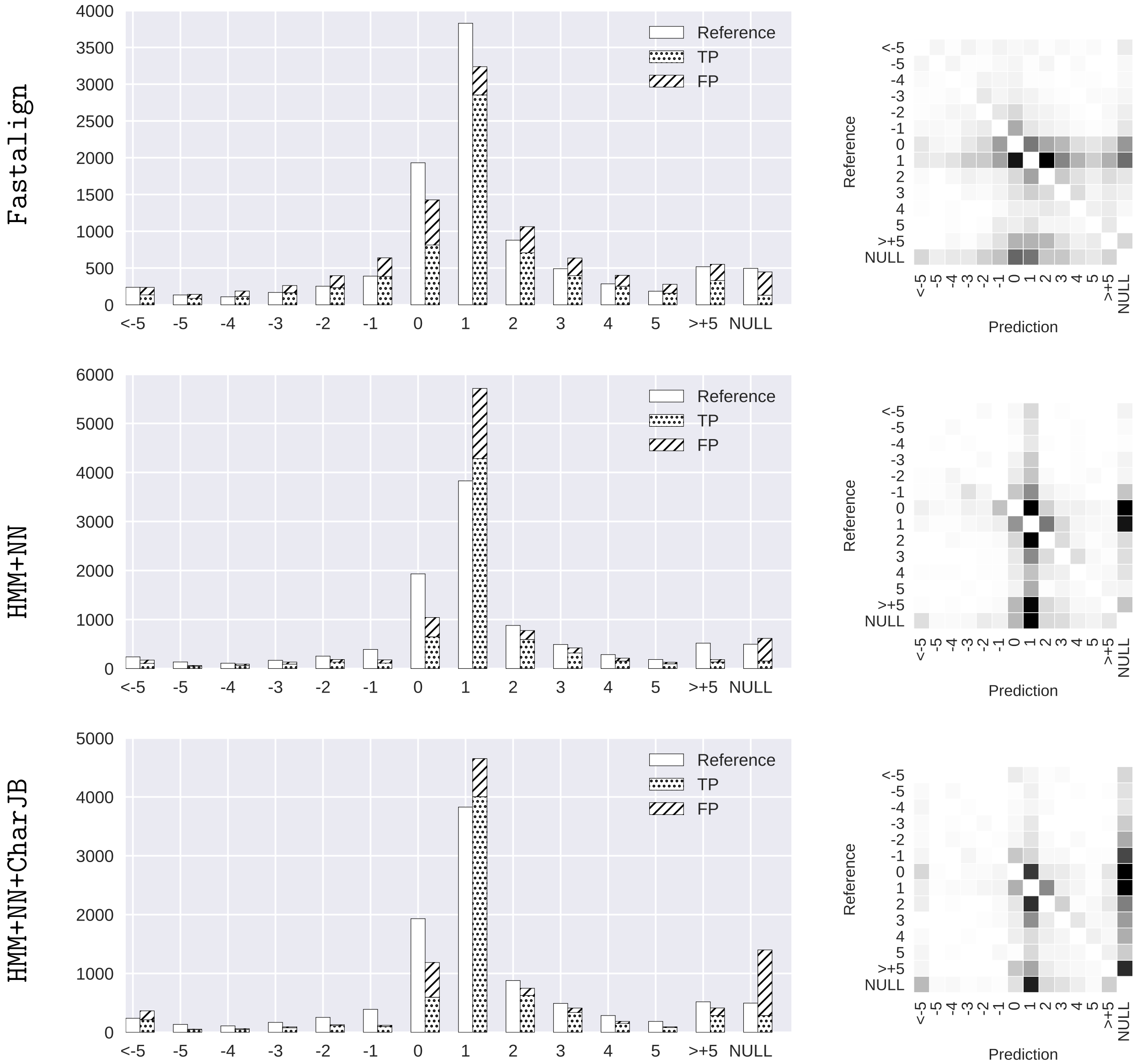


Figure 4.16: Distortion distribution for the direction English-German: Correct (TP) and incorrect (FP) jump widths for source words on the left graph. Confusion matrices on the right graph: The darker cell, the greater the number of confusions. `Fastalign`: In the left graph, `Fastalign` generates about 400 incorrect jumps of length 1, which is much smaller than the corresponding number of `HMM+NN` (about 1500 jumps). In the right graph, `Fastalign` confuses the jumps of length 0 and 1 with the longer jumps. `HMM+NN`: It generates too many short jumps equal to 1 (about 1500 jumps), as well as too many null alignments (about 600 links), as can be seen in the left graph. In the right graph, most longer jumps are confused with the short jumps. Moreover, a number of short jumps in reference become jumps to NULL token in prediction. `HMM+NN+CharJB`: In the left graph, for jumps of length 1, it generates less incorrect jumps (about 600 incorrect jumps) than `HMM+NN` and more correct jumps than `Fastalign`. We can see that not only short jumps in reference become jumps to NULL token in prediction.

4.7.5 One-to-one and many-to-one links

We evaluate the performance with varying alignment types: one-to-one and one-to-many. Figure 4.17 shows predicted alignments for English-Romanian. All `HMM` models encourage one-to-one alignments, which produces most of the correct links. This is clearly because one-to-one is the most frequent link type among the four alignment types. We compare the variants of `HMM` to find the models that capture best many-to-one of alignments.

- English on source side: There is a small number of many-to-one links in the case of French, Japanese and Vietnamese; a large corresponding number in the case of German, Romanian and Czech. The neural distortion models generate a smaller number of many-to-one links than other models, which seems to correspond well to the pattern observed in French, Japanese and Vietnamese. The character-based models are a good choice for German and Romanian where they predict many more many-to-one links. `+NNCtxCNN` is the best option for Czech.

- Foreign language on source side: We expect the opposite behavior where there are a large number of many-to-one links in French, Japanese and Vietnamese, and a small number in German, Romanian and Czech. The character-based models capture well the pattern for French. The neural distortion model accomplishes well this task in the cases of German, Romanian, Japaneses and Vietnamese. For Czech, the contextual models are also a good approach.

In short, the neural distortion models and character-based translation models prove their usefulness in the recognition of alignment patterns for all languages observed except Czech. The contextual models seem to be an appropriate approach for this language.

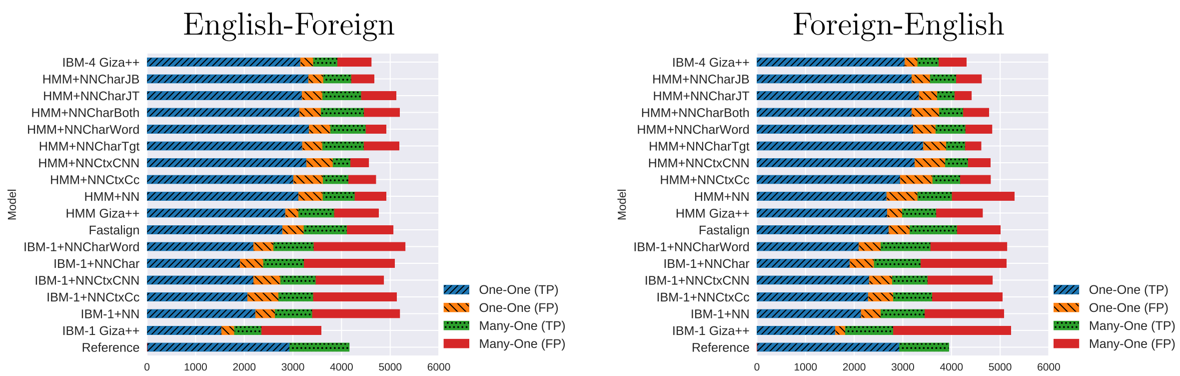


Figure 4.17: Results of our neural models: Alignment types for English-Romanian (both directions)

We observe one-to-many and many-to-many links for English-German in the case of attention-based models (Figure 4.18). Complete results are in [Ngo Ho, 2021, Appendix C.4]. Using the same threshold, we see that alignment results from the model `GT` consist of a smaller number of many-to-many links than the model `UT`. Note that these many-to-many links do not greatly improve the performance of attention-based models.

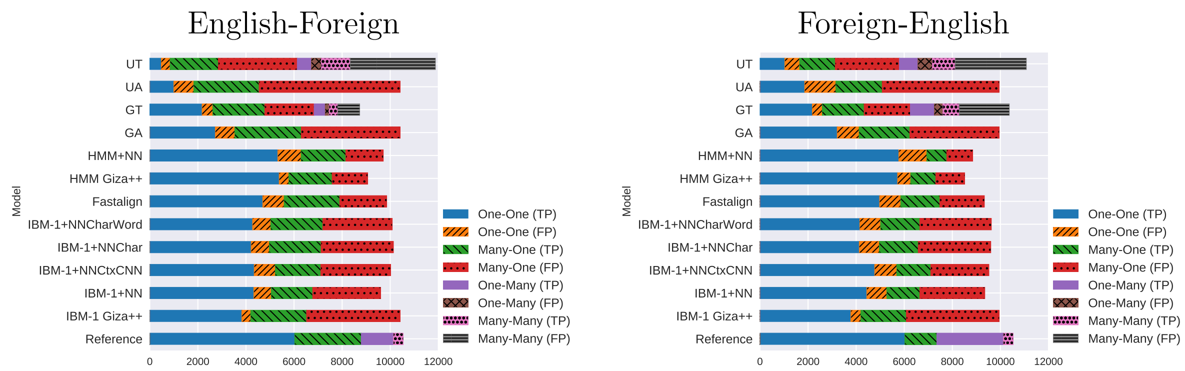


Figure 4.18: Results of our attention-based models: Alignment types for English-German (both directions)

4.7.6 Do neural network models have a problem with rare/unknown words?

Complete results for rare words and unknown words are in [Ngo Ho, 2021, Appendix B.6] and [Ngo Ho, 2021, Appendix B.7] respectively. We consider the effects of neural models in solving garbage collector problem by observing the fertility of rare words. As can be seen in Table 4.10 (English-Czech), we see the clear benefits of using neural translation models: both `IBM-1` variants and `HMM` variants yield a clear reduction of fertility. We notice that this reduction yields a loss in recall with a large number of null links. This means that our NN models only keep sure links involving the rare words. We also see a better accuracy and a better precision which contributes to a F-score improvement.

- For `IBM-1` variants, the loss in recall is substantial (about -50%). This causes a bad effect on F-score.
- For `HMM` variants, the loss in recall is much smaller than for `IBM-1` variants, yielding a better F-score. The only explanation is that distortion models play a key role in this improvement.

The improvements for rare words are also illustrated by an example of a Romanian rare word "sireturi", which is misaligned by `IBM-1` to common English words such as "must", "generate", "such", "low", "-" and "down". When using `IBM-1+NN`, "sireturi" is misaligned only to "demoiselle" (Figure 4.19).

For unknown words, we report performance for the case where the neural models and the baselines use the same vocabulary size for known words. Therefore, we consider two cases:

- 50K word vocabulary: For the baselines, we replace all words that are not the top 50K most frequent with the UNK token. We compare these baselines with the neural models only using word embeddings. As can be seen in Table 4.11 (English-Czech), we see the clear benefits of using neural translation models that they create a great improvement in both precision and recall, yielding a better F-score. Similar benefit is found in other language pairs/for both directions except for the direction French-English. In this case, `HMM+NN` and `HMM+NN+Cc` still lag a few points behind `HMM Giza++`.
- Full vocabulary: The baselines in this case do not need the UNK token to cover unknown words because their training and test corpus are concatenated (Section 3.7). We compare them with our character-based where we remove the effect of unknown target words. Note that `NNCharBoth` totally eliminates unknown words whereas other models still suffer 50K word vocabulary on the source side.

  In general, alignment for unknown words are clearly improved by our neural models except for the English-Czech language pair. We report the model performance for this worse case

Models	English						Foreign					
	#	FE	ACC	PRE	REC	F1	#	FE	ACC	PRE	REC	F1
`IBM-1 Giza++`	1961	4.25	85.54	15.96	56.09	*24.85*	3365	2.86	90.68	23.6	46.06	*31.2*
`IBM-1+NN`	582	1.26	93.19	21.31	22.22	21.75	1131	0.96	93.55	19.01	12.47	15.06
`IBM-1+NNCtxCc`	709	1.54	92.38	19.04	24.19	21.31	1458	1.24	92.62	13.99	11.83	12.82
`IBM-1+NNCtxCNN`	572	1.24	92.96	18.18	18.64	18.41	1362	1.16	93.01	16.81	13.28	14.84
`IBM-1+NNChar`	637	1.38	92.98	21.66	24.73	**23.1**	1817	1.55	93.54	30.6	32.25	31.4
`IBM-1+NNCharWord`	767	1.66	92.08	18.77	25.81	21.74	1596	1.36	93.99	33.21	30.74	**31.93**
`Fastalign`	700	1.52	95.94	51.86	65.05	*57.71*	1489	1.27	95.84	55.41	47.85	*51.35*
`HMM Giza++`	1623	3.52	89.42	24.52	71.33	36.5	2878	2.45	93.61	38.26	63.86	47.85
`HMM+NN`	521	1.13	96.63	61.23	57.17	59.13	1409	1.2	96.32	62.17	50.81	55.92
`HMM+NNCtxCc`	434	0.94	96.85	66.82	51.97	58.47	1142	0.97	96.16	62.35	41.3	49.69
`HMM+NNCtxCNN`	458	0.99	97.17	70.52	57.89	63.58	1408	1.2	96.23	60.94	49.77	54.79
`HMM+NNCharTgt`	461	1	97.14	69.85	57.71	63.2	1205	1.02	97.23	78.42	54.81	64.53
`HMM+NNCharWord`	512	1.11	97.11	67.58	62.01	**64.67**	1257	1.07	97.46	80.67	58.82	**68.03**
`HMM+NNCharBoth`	422	0.92	97.02	69.91	52.87	60.2	1176	1	97.33	80.61	54.99	65.38
`HMM+NNCharJT`	428	0.93	96.96	68.69	52.69	59.63	1044	0.89	97.13	80.94	49.01	61.05
`HMM+NNCharJB`	520	1.13	96.2	55.77	51.97	53.8	1272	1.08	97.17	75.94	56.03	64.49
`IBM-4 Giza++`	1468	3.18	90.83	28.13	74.01	*40.77*	2430	2.07	95	46.79	65.95	*54.74*

Table 4.10: Models for English-Czech: # links, fertility (FE), accuracy (ACC), F-score (F1), precision (PRE) and recall (REC) for the rare target words in the direction Czech-English and in the direction English-Czech

in Table 4.12. We do not see the improvement for the variants of `IBM-1`. For the `HMM` variants, in the direction Czech-English, the benefit of using character-based models is less clear while only `HMM+NNCharWord` beats `Fastalign`. Moreover, `NNCharBoth` fails to improve more than other character-based models. We notice that the failure comes from a large loss in recall, which again highlights the problem of unaligned words.

For both cases, the largest gain is often found in the directions where there are more unknown words in the target side than the source side.

Models	English						Foreign					
	#	FE	ACC	PRE	REC	F1	#	FE	ACC	PRE	REC	F1
`IBM-1 Giza++`	2059	1.09	92.62	13.4	11.26	*12.24*	3630	0.7	93.84	11.49	5.57	*7.5*
`IBM-1+NN`	2433	1.29	92.83	21.33	21.18	21.25	5318	1.03	93.59	19.76	14.04	16.42
`IBM-1+NNCtxCc`	2925	1.56	92	18.56	22.15	20.2	6595	1.28	92.84	16.03	14.12	15.01
`IBM-1+NNCtxCNN`	2530	1.35	92.44	18.3	18.89	18.59	6596	1.28	92.95	17.5	15.41	16.39
`Fastalign`	2258	1.2	95.17	46.99	43.29	45.06	4335	0.84	95.28	45.49	26.34	*33.36*
`HMM Giza++`	1642	0.87	95.85	56.82	38.07	*45.59*	2174	0.42	95.65	54.92	15.95	24.72
`HMM+NN`	2298	1.22	96.28	59.97	56.22	58.03	6313	1.22	96.24	59.51	50.18	54.45
`HMM+NNCtxCc`	1867	0.99	96.5	65.4	49.82	56.55	5141	0.99	96.15	60.34	41.43	49.13
`HMM+NNCtxCNN`	2012	1.07	96.61	65.71	53.94	59.24	6456	1.25	96.24	59.37	51.2	54.98
`IBM-4 Giza++`	1251	0.67	95.46	50.6	25.83	34.2	1239	0.24	95.71	62.79	10.39	17.83

Table 4.11: Models for English-Czech: # links, fertility (FE), accuracy (ACC), F-score (F1), precision (PRE) and recall (REC) for unknown target words in the direction Czech-English and in the direction English-Czech. Note that the training data for all models including the baselines only has a vocabulary containing the most frequent 50K words.

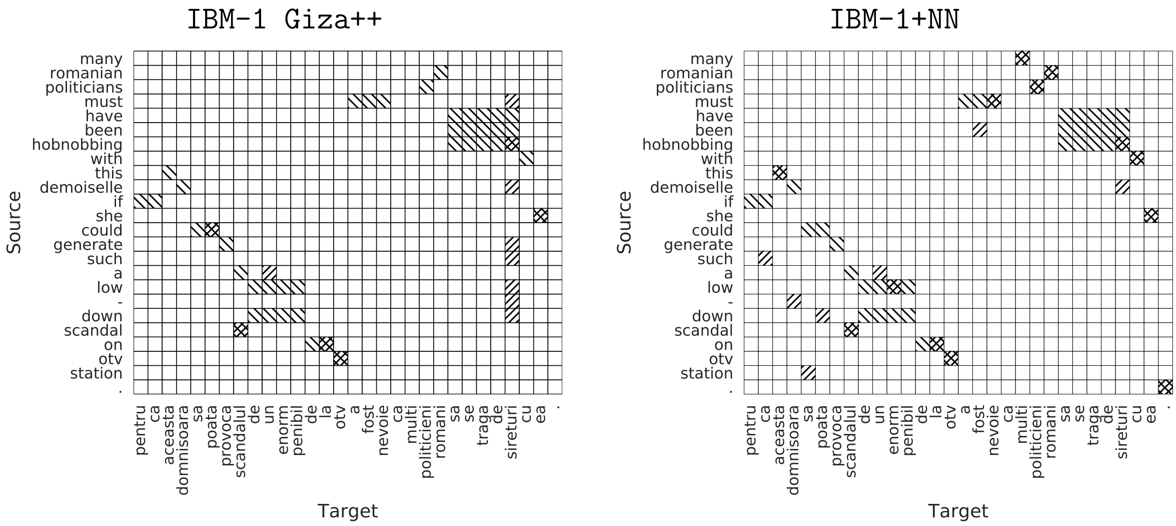


Figure 4.19: Example of alignment links for a Romanian rare word "sireturi". Back diagonal hatching, diagonal hatching and crossed diagonal hatching represent a reference alignment link, a predicted alignment link and a correctly predicted alignment link by `IBM-1 Giza++` and `IBM-1+NN`. We see that this Romanian word is misaligned by `IBM-1 Giza++` to common English words such as "must", "generate", "such", "low", "-" and "down". When using `IBM-1+NN`, "sireturi" is misaligned only to "demoiselle"

Models	English						Foreign					
	#	FE	ACC	PRE	REC	F1	#	FE	ACC	PRE	REC	F1
`IBM-1 Giza++`	6931	4.33	85.4	16.87	55.53	*25.87*	8487	3.33	89.86	20.9	48.91	*29.29*
`IBM-1+NNChar`	2210	1.38	92.9	23.94	25.13	24.52	4393	1.73	92.9	23.06	27.93	25.26
`IBM-1+NNCharWord`	2761	1.73	91.85	20.39	26.75	23.14	3690	1.45	93.68	26.8	27.27	27.03
`Fastalign`	2118	1.32	96.29	59.54	59.9	*59.72*	3056	1.2	96.22	57.04	48.06	*52.16*
`HMM Giza++`	5702	3.57	89.75	27.24	73.78	39.78	7488	2.94	92.59	32.41	66.91	43.67
`HMM+NNCharTgt`	1665	1.04	96.49	64.86	51.31	57.29	2489	0.98	97.18	74.97	51.45	61.02
`HMM+NNCharWord`	1853	1.16	96.59	64.6	56.86	**60.49**	2964	1.16	97.24	71.9	58.75	**64.66**
`HMM+NNCharBoth`	1635	1.02	96.24	61.59	47.84	53.85	2611	1.03	97.15	73.42	52.85	61.46
`HMM+NNCharJT`	1533	0.96	96.32	63.54	46.27	53.55	2130	0.84	96.97	74.98	44.03	55.48
`HMM+NNCharJB`	1931	1.21	95.91	55.88	51.26	53.47	3016	1.18	96.82	65.58	54.54	59.55
`IBM-4 Giza++`	5132	3.21	91.11	30.79	75.06	43.66	6058	2.38	94.39	40.82	68.18	51.07

Table 4.12: Models for English-Czech: # links, fertility (FE), accuracy (ACC), F-score (F1), precision (PRE) and recall (REC) for the unknown target words in Czech-English and in English-Czech. Note that there is no unknown words in the training data for the baselines.

4.7.7 Issues with function/content words

We analyze the links errors by two main categories: function and content words (Section 3.8). Complete results are in [Ngo Ho, 2021, Appendix B.8]. Regarding the top graphs in Figure 4.20, the main observation is that content words benefit from neural network models whereas the errors for function words are almost unchanged. The most important gain is obtained with character-based models. Similar trends are found in other language pairs/directions.

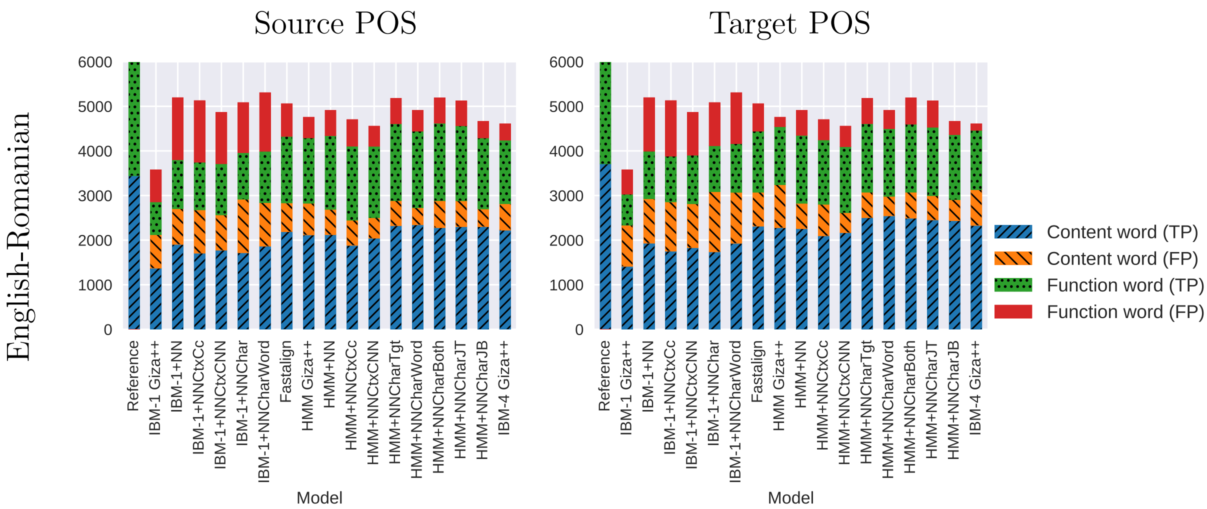


Figure 4.20: PoS results for the direction English-Romanian: The number of target words that align with a content/function source word (left graph). The number of source words that align with a content/function target words (right graph).

4.7.8 Does symmetrization still improve alignments ?

We obtain symmetrized alignments[4] that greatly outperform their corresponding baselines. Complete results are in [Ngo Ho, 2021, Appendix B.9.2]. The gain could be as large as -5/6 AER for English-Czech/Romanian, and more than -10 AER for English-Japanese/Vietnamese. We observe the case of English-French (Table 4.13). Our best results outperform `Giza++ IBM-4`. We note that `HMM+NNCharTgt`, which outperforms `IBM-4` for both directions, is worst after symmetrization. This is because `IBM-4` has a smaller recall, but a higher precision, in both directions. As the symmetrization heuristic selects links that are predicted in both directions [Koehn et al., 2005], it yields an improved prevision without impacting the recall for `IBM-4`. This loss in the recall is also found in our NN distortion models.

Even better scores are obtained when symmetrization uses the best model in each direction (Table 4.14): doing so in English-Romanian with our best HMM models brings us an additional improvement of about +1 AER.

Models	English-Foreign		Foreign-English		GDF			
	AER	F1	AER	F1	AER	F1	PRE	REC
`HMM+NNCharTgt`	9.17	47.22	9.56	47.87	8.41	48.99	90.12	33.64
`HMM+NNCharWord`	10.45	47.33	10.27	48.56	9.33	49.64	88.61	34.48
`HMM+NNCharBoth`	10.9	46.74	11.17	47.5	10.51	48.51	87.16	33.6
`HMM+NNCharJT`	8.41	44.71	7.7	44.45	6.26	45.39	94.55	29.87
`HMM+NNCharJB`	8.47	44.38	7.74	46.26	6.81	45.83	93.32	30.38
`IBM-4 Giza++`	10	44.43	9.64	45.43	7.03	46.32	93.55	30.78

Table 4.13: Grow-diag-final: Alignment error rate (AER), F-score (F1) for English-French. Our best results outperform `IBM-4 Giza++`.

For attention-based models, we observe similar trends when using GDF, e.g., a gain of -4 AER for `GT` in Table 4.15. Compared with our neural variant `IBM-1+NN` and the baseline `IBM-1 Giza++`, the model `GT` obtains the largest recall of 29.29 points, yielding the best F-score. This is because directional attention-based alignments contain many-to-many links and GDF benefits from them. Another explanation is that English-French has a large number of many-to-many

[4]Using the grow-diag-final heuristic proposed in Koehn et al. [2005].

Models	IBM-1				HMM			
	AER	F1	PRE	REC	AER	F1	PRE	REC
English-French	17.86	40.83	81.54	27.23	6.26	45.39	94.55	29.87
English-German	27.54	69.61	78.14	62.76	22.42	74.71	84.83	66.74
English-Romanian	38.13	61.88	79.03	50.85	24.95	75.07	81.48	69.59
English-Czech	29.22	57.83	76.86	46.35	17.53	68.98	83.63	58.69
English-Japanese	44.79	55.21	50.41	61.01	25.28	74.72	73.28	76.22
English-Vietnamese	43.61	56.4	94.49	40.2	25.32	74.69	93.47	62.19

Table 4.14: Grow-diag-final for the best models in each direction: Alignment error rate (AER), F-score (F1), precision (PRE) and recall (REC).

links and possible reference links. Complete results are shown in [Ngo Ho, 2021, Appendix C.6.2].

Direction	AER	F1	PRE	REC
`IBM-1 Giza++`				
English-Foreign	40.1	26.7	71.55	16.41
Foreign-English	33.9	36.49	59.24	26.37
GDF	25.19	33.83	82.75	21.26
`IBM-1+NN`				
English-Foreign	27.96	36.42	69.66	24.65
Foreign-English	27.21	38.08	68.12	26.43
GDF	*17.86*	39.48	*82.89*	25.91
Attention-based `GT`				
English-Foreign	35.63	37.2	66.85	25.77
Foreign-English	34.88	40.18	67.35	28.63
GDF	31	**41.48**	71.05	**29.29**

Table 4.15: Alignment error rate (AER), F-score (F1), precision (PRE) and recall (REC) for English-French in both directions and for GDF.

4.7.9 Is more data usually better ?

In order to find a way of improving more the performance of our existing models, we revisit here the case of German where our neural models obtain the smallest gain. In detail, we try to understand why our models fail to greatly increase the model performance and observe the behaviors of several neural models when increasing the training corpus size.

Alignment errors of our neural translation models: For this language pair, our neural models cannot outperform their discrete counterparts, except the two models using neural distortion models. As can be seen in Table 4.16, they obtain better AER scores than `HMM Giza++` because they predict fewer alignment links (favoring precision over recall). The same strategy is used by `IBM-4 Giza++`. This explains a large number of unaligned source words (Figure 4.21) and incorrect jumps to NULL token (Figure 4.16).

This loss of recall is also observed for rare words. In Figure 4.22, for the rare German word “hochgelegen", all correct links are found by `HMM Giza++` whereas this German word is also misaligned to common English words “helping", “where", “very", “up" and “list". In contrast, the model `HMM+NN+CharJB` correctly aligns this rare word with only one English word “high"

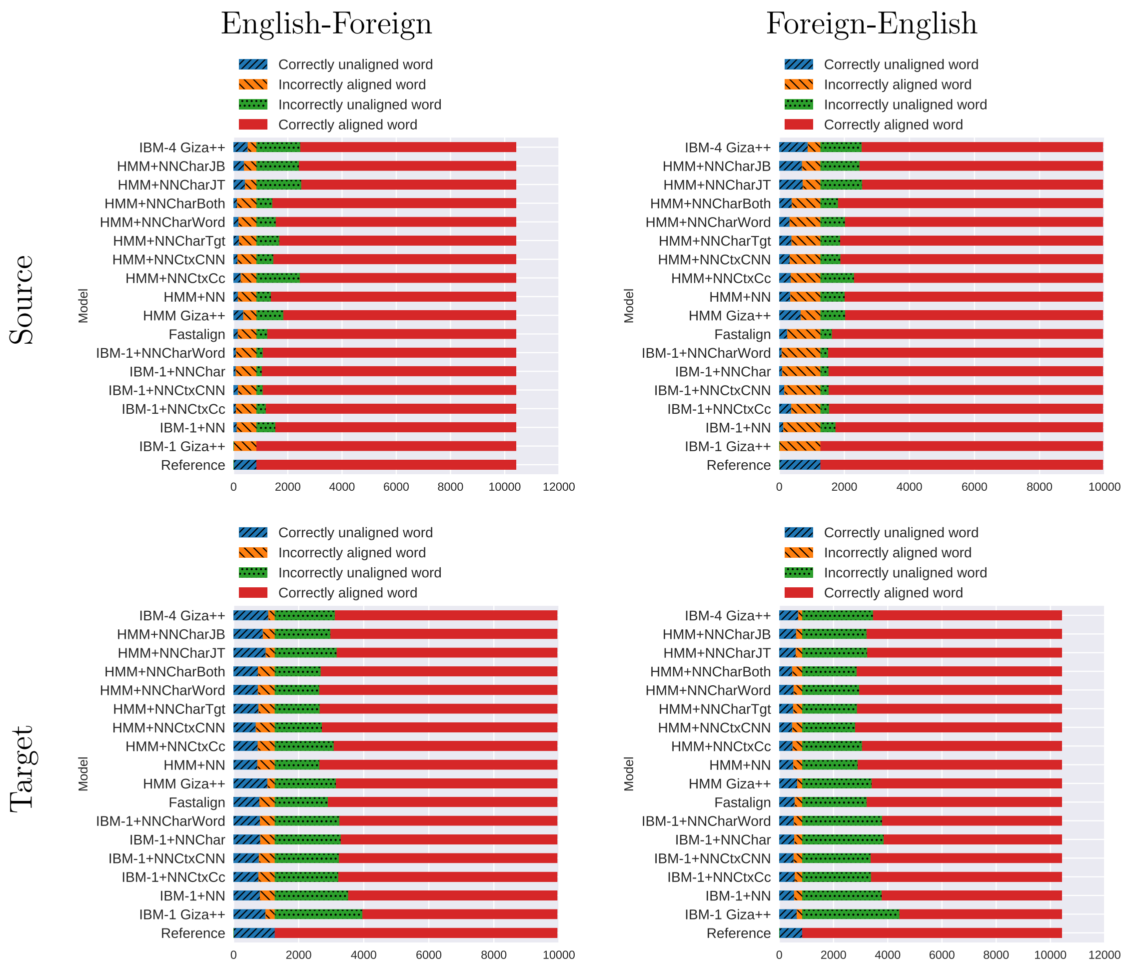


Figure 4.21: Results of our neural models: Unaligned words for English-German

(Note that “hochgelegen”-“high” is a sure link). The English word “very" and “up" are aligned to the NULL token.

Increasing the training corpus size One solution that should be considered here is to increase the number of sentences in the training corpus. We use a clean parallel corpus Paracrawl[5] produced by Pham et al. [2018]. Note that our default training corpus shown in Section 3.1.1 consists of ∼1.5M sentences. We add ∼1.5M sentences and then ∼4.5M sentences to obtain two corpora of 3M sentences and 6M sentences. We train these corpora with the models `HMM+NN` (`HMM+NN+3M` and `HMM+NN+6M`), `HMM+NNCharTgt` (`HMM+NNCharTgt+3M` and `HMM+NNCharTgt+6M`) and `HMM+NNCharJB` (`HMM+NNCharJB+3M` and `HMM+NNCharJB+6M`).

We clearly see the benefits when using a very large corpus. In fact, our vanilla models `HMM+NN` gain 2 points for both AER and F-score. The character-based models surpass their counterpart `HMM Giza++`. The models `HMM+NNCharJB` with neuralized distortion models also outperform `IBM-4 Giza++`. The larger corpus helps to gain -1 AER and +1 F-score. We recognize that the improvements are found in both precision and recall. This clearly means that using a very large corpus does help to generate better word embeddings.

We observe the similar benefits for (less frequent) words that occur maximum 50 times in our default training (Table 4.17). We use a higher threshold 50 times instead of once, helping to more clearly display these benefits. As can be seen in this table, a very large corpus improves recall and reduces the over-generation of NULL links, but still keeps a good fertility rate. We also show a nice example of how neural models correct alignment errors and how a large corpus continues to correct the rest of the errors in Figure 4.23. Similar results are found for unknown words [Ngo Ho, 2021, Appendix B.7.2].

[5] http://paracrawl.eu/

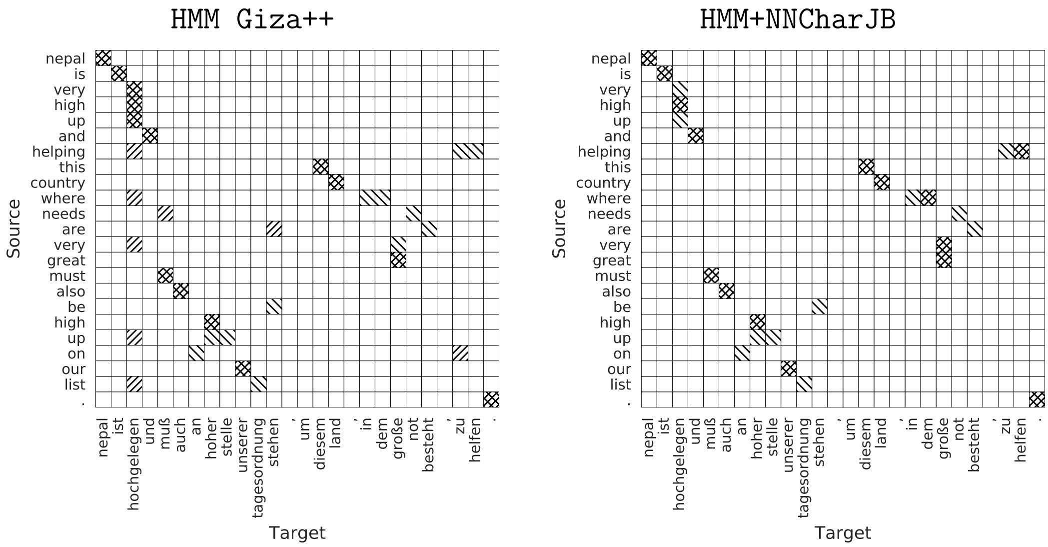


Figure 4.22: Example of German rare word "hochgelegen": Sure links are "hochgelegen"-"high" and "hochgelegen"-"up", possible link is "hochgelegen"-"very". Back diagonal hatching, diagonal hatching and crossed diagonal hatching represent a reference alignment link, a predicted alignment link and a correctly predicted alignment link.

Models	English-Foreign					Foreign-English				
	AER	F1	PRE	REC	ACC	AER	F1	PRE	REC	ACC
`Fastalign`	28.98	68.75	71.11	66.54	97.35	31.28	66.47	70.73	62.69	97.23
`HMM Giza++`	23.92	73.3	79.23	68.2	97.82	26.33	71.04	79.47	64.23	97.7
`HMM+NN`	26.78	70.95	73.94	68.2	97.55	29.44	68.21	74.69	62.76	97.44
`HMM+NNCharTgt`	26.04	71.57	75.99	67.64	97.64	28.11	69.48	75.59	**64.29**	97.52
`HMM+NNCharWord`	24.98	72.64	76.53	**69.13**	97.72	29.77	67.76	74.12	62.4	97.4
`HMM+NNCharBoth`	27.14	70.6	73.65	67.79	97.52	29.31	68.34	74.11	63.41	97.42
`HMM+NNCharJT`	23.79	73.15	**82.8**	65.52	**97.89**	25.21	71.85	**83.64**	62.98	97.84
`HMM+NNCharJB`	**23.69**	**73.38**	82.38	66.15	97.9	**24.9**	**72.16**	83.36	63.61	**97.85**
`HMM+NN+3M`	25.19	72.55	76.12	69.31	97.7	27.95	69.67	76.83	63.73	97.57
`HMM+NN+6M`	24.79	73.04	76.25	70.08	97.73	26.71	71.03	78.32	64.98	97.68
`HMM+NNCharTgt+3M`	23.51	74.15	79.32	69.62	97.87	26	71.65	78.66	65.78	97.72
`HMM+NNCharTgt+6M`	22.67	74.97	80.25	**70.35**	97.94	24.88	72.87	79.87	**66.99**	97.81
`HMM+NNCharJB+3M`	20.1	77.02	87.03	69.08	98.19	21.35	75.83	88.54	66.31	98.15
`HMM+NNCharJB+6M`	**19.99**	**77.16**	**87.2**	69.2	**98.2**	**20.84**	**76.39**	**89.52**	66.61	**98.19**
`IBM-4 Giza++`	21.46	75.48	85.79	67.39	98.08	23.31	73.63	86.56	64.06	97.99

Table 4.16: Alignment error rate (AER), accuracy (ACC), F-score (F1), precision (PRE) and recall (REC) for English-German. The bottom part of the table report scores with increased training data (3M, then 6M).

Models	English						Foreign					
	#	FE	ACC	PRE	REC	F1	#	FE	ACC	PRE	REC	F1
`Fastalign`	162	1.14	96.19	71.6	62.7	66.86	430	1.3	96.83	82.79	65.93	73.4
`HMM Giza++`	388	2.73	90.96	38.66	81.08	52.36	864	2.62	93.39	50.12	80.19	61.68
`HMM+NN`	150	1.06	95.26	64	51.89	57.31	462	1.4	96.19	74.89	64.07	69.06
`HMM+NNCharTgt`	135	0.95	96.62	80.74	58.92	68.12	432	1.31	96.95	83.8	67.04	74.49
`HMM+NNCharJB`	138	0.97	96.66	80.43	60	68.73	424	1.28	97.03	85.14	66.85	74.9
`HMM+NN+3M`	147	1.04	95.56	67.35	53.51	59.64	460	1.39	96.46	77.39	65.93	71.2
`HMM+NN+6M`	151	1.06	95.7	68.21	55.68	61.31	469	1.42	96.57	77.83	67.59	72.35
`HMM+NNCharTgt+3M`	138	0.97	96.85	82.61	61.62	70.59	437	1.32	97.21	85.81	69.44	76.77
`HMM+NNCharTgt+6M`	131	0.92	97.15	87.79	62.16	72.78	440	1.33	97.25	85.91	70	77.14
`HMM+NNCharJB+3M`	134	0.94	97.05	85.82	62.16	72.1	420	1.27	97.37	88.81	69.07	77.71
`HMM+NNCharJB+6M`	141	0.99	97.28	86.52	65.95	74.85	425	1.29	97.48	89.41	70.37	78.76
`IBM-4 Giza++`	337	2.37	92.38	43.32	78.92	55.94	757	2.29	94.66	56.94	79.81	66.46

Table 4.17: # links, fertility (FE), accuracy (ACC), F-score (F1), precision (PRE) and recall (REC) for the rare target words in the direction German-English and in the direction English-German. The bottom part of the table report scores with increased training data (3M, then 6M). Note that in this table a word is rare if it occurs less 50 times in our training corpus.

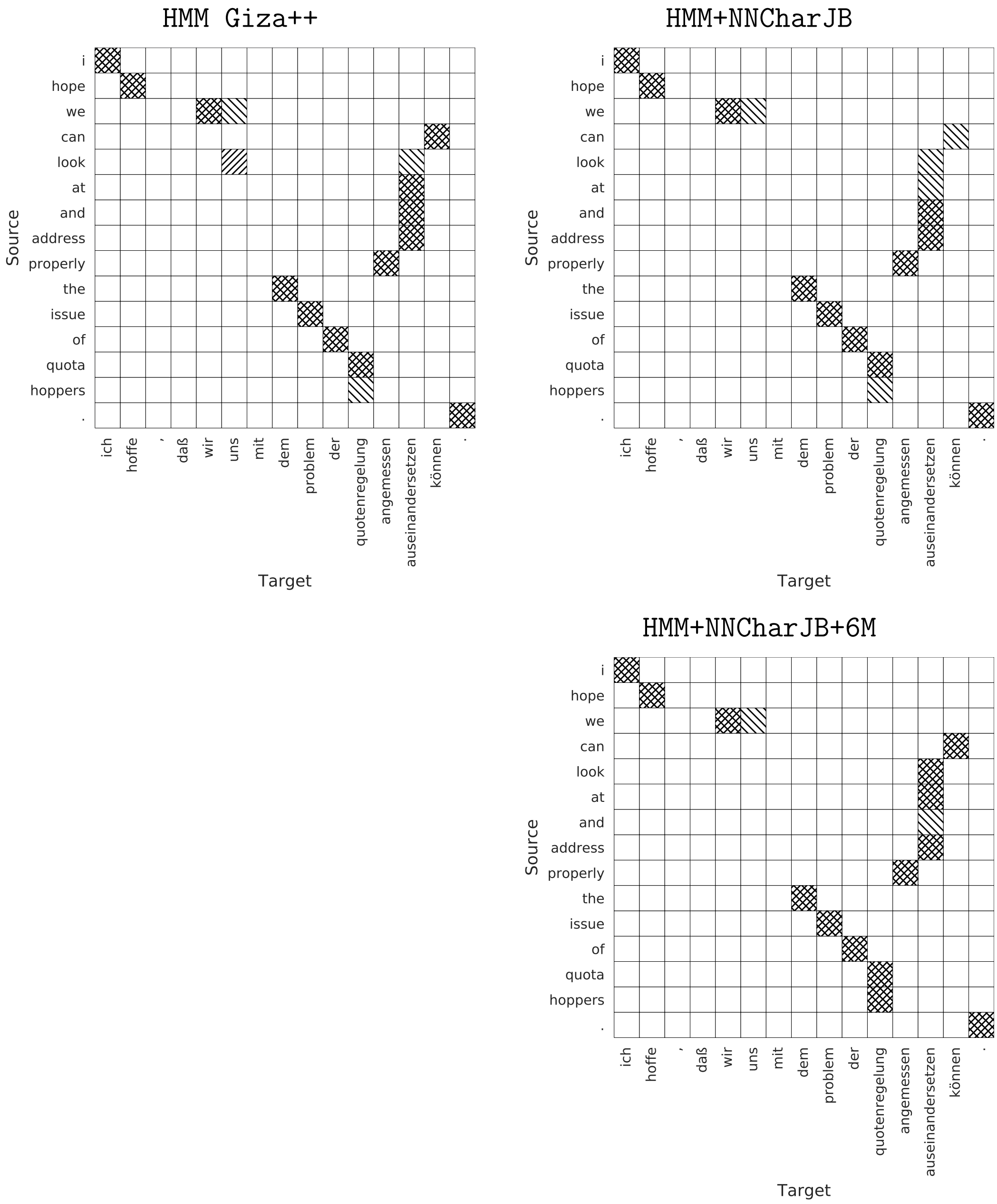


Figure 4.23: Example of German word "auseinandersetzen": We see how a neural model (`HMM+NNCharJB`) corrects alignment errors of the discrete model `HMM Giza++` and how a large training corpus helps to correct unaligned words. This word occurs 453 times in our default training corpus. Note that back diagonal hatching, diagonal hatching and crossed diagonal hatching represent a reference alignment link, a predicted alignment link and a correctly predicted alignment link

4.8 Summary

In this chapter, we described artificial neural networks (Section 4.1) and their applications in NLP. In detail, we presented word embeddings (Section 4.1.1) and two common neural network architectures: Convolutional neural networks (Section 4.1.2), (bidirectional) recurrent neural networks with long short-term memory (Section 4.1.3). We surveyed the works related to neural word alignment models in Section 4.2. We replaced the traditional count-based translation models with several variants of neural networks, notably contextual models and character-based models (Section 4.3). We neuralized the distortion models in Section 4.4 using character-based representations. Details of our training algorithm and our experiments are respectively in Section 4.5 and Section 4.6. In Section 4.7, we observed the performance of our models in word alignment for six language pairs (English with French, German, Czech, Romanian, Japanese and Vietnamese) and discussed how neural network overcomes alignment difficulties of `Giza++` and `Fastalign`.

One important observation from our experiments is that neural models can help achieve remarkable improvements in AER scores for most language pairs, with the higher gains observed for Czech and Romanian, two morphologically rich languages, in a small data condition. We also showed that most of these gains are due to a decrease in non-null link errors. Moreover, NN models yield a clear benefit for long sentences. Content words benefit from these models whereas the errors for function words are almost unchanged. Note that using a larger training corpus helps to gain more performance points (Section 4.7.9). We summarize some of our major findings for each type of model as follows.

- `+NN`: Most of the performance improvement is already achieved by this vanilla NN model.

- `+NNCtx`: The difference between the models using concatenation (`+NNCtxCc`) and convolutions (`+NNCtxCNN`) is limited. The latter approach seems to be on average the best choice. The largest gains are observed in small data conditions (Romanian-English, Czech-English, Japanese-English and Vietnamese-English) when English is on the target side. Shortly, the context helps to disambiguate alignment links for English words by improving the translation distribution.

- `+NNChar`: One obvious benefit is that character-based representations help to differentiate the translation model for rare words. Models using character-based in the target yield significant and consistent gains, especially also in small data conditions. We saw that is that the pure character-based approach (`+NNCharTgt`) should be preferred given a sufficiently large dataset (English-French/German) when this is not the case, word information (`+NNCharWord`), which is easier to train (i.e., using a simpler architecture), can also prove helpful.

- `+NNChar` with the neuralized distortion models: The models `+NNCharJT` and `+NNCharJB` gain some more points compared with their character-based counterparts. Moreover, neural distortion models over-predict null links, which yields a large number of correctly unaligned words. This can be helpful for Vietnamese and Japanese where there are a large number of unaligned words.

- Attention-based models: The model `G` (Generate first) shows slight improvements compared with IBM-1.

To the best of our knowledge, our best results are the strong models compared with other published numbers [Ngo Ho, 2021, Appendix F] for English vs French, German, Romanian and Japanese. For English-French, our best models outperform the models of Kamigaito et al. [2014], Legrand et al. [2016], Rios et al. [2018], Zenkel et al. [2019], Ding et al. [2019b], Nagata et al. [2020]. We see a small improvement of about -1 AER for English-German[6] and English-Romanian. For English-Japanese, our models can reach 24.92 AER, better than the models of Kondo et al. [2013] and Kamigaito et al. [2014].

Our analysis also suggests that the alignment problem is still far from solved, and that progress still needs to be made in the three issues:

- Prediction of null words: In our model implementation except for NN+CtxCc, null is simply one special word in the vocabulary, which does not encode information of the target word that it replaces. We therefore need a better approach to process this token.
- Towards symmetric models: Our neural models are asymmetrical and use heuristic post-process (e.g. GDF) to obtain symmetrical alignments. We will discuss how to generate many-to-many links using subwords in Chapter 6.
- More fine-grained prediction requiring better word representations on the target side: One remarkable solution is variational autoencoders which helps to improve word representations via the reconstruction process. We will present this approach in Chapter 5.

Moreover, we also notice that the training time for the neural network systems is much longer than for the baselines.

[6]This is the case where we do not use extra bilingual corpus.

Chapter 5

Generative latent neural alignment models

A variational autoencoder (VAE), a generative model, aims to represent high-dimensional complex data via a low-dimensional latent space. This model is proposed by Kingma and Welling [2014], Rezende et al. [2014]. In VAEs, we can model priors on the latent variables, which helps to control latent representations and show promise in generating many kinds of complicated data. Note that the assumptions of these models are weak and training is fast via back-propagation. They do make an approximation, but the error introduced by this approximation is arguably small given high-capacity models [Doersch, 2016, Cho, 2014, Girin et al., 2020]. VAEs are used in a host of applications such as image modeling [Pu et al., 2016, Higgins et al., 2017, Gulrajani et al., 2017], language modeling [Bowman et al., 2016, Miao et al., 2016], machine translation [Eikema and Aziz, 2018, Deng et al., 2018, Pagnoni et al., 2018, Su et al., 2018, Zhang et al., 2016], syntactic parsing [Corro and Titov, 2019], labeled sequence transduction [Zhou and Neubig, 2017], speech modeling and handwriting generation [Chung et al., 2015].

Our main source of inspiration is the model of Rios et al. [2018] to approach the unsupervised estimation of neural alignment models. They exploit neural versions of conventional alignment (IBM-1/2) models, intending to improve word representations in low resource contexts. We revisit here this model, trying to analyze the reasons for its unsatisfactory performance and we extend it in several ways, taking advantage of its fully generative nature.

- We generalize the approach, initially devised for IBM model 1 [Rios et al., 2018], to the HMM model by introducing Markovian dependencies.
- We propose a sharing parameter approach which highlights the symmetric nature of the problem.
- We explore ways to effectively enforce symmetry constraints.
- We study how these models could benefit from monolingual data.

We first describe variational autoencoders in Section 5.1 and a fully generative model of word alignments in Section 5.2.1. We then introduce our HMM variational model in Section 5.2.2. To make our models more symmetric, we propose a sharing parameter approach in Section 5.2.3. We also present a way to enforce the agreement in alignment (Section 5.2.4). We discuss how monolingual data can help to improve alignment performance in Section 5.2.5. We show our experiments in Section 5.3 and finally evaluate our word alignment variational models in Section 5.4. A shorter version of this work is published in Ngo Ho and Yvon [2020].

Contents

5.1 Variational auto-encoders

An autoencoder (AE) neural network is an unsupervised learning algorithm setting the target values to be equal to the inputs [Goodfellow et al., 2016]. The main role of this neural network is to discover the inner structure of the data by defining the constraints on the network, e.g., limiting the number of hidden units. In other words, it tries to reproduce a representation or a different form of input. Latent variable models are a class of statistical models that seek to model the relationship of observed variables with a set of unobserved, latent variables, and can allow for the modeling of more complex, generative processes. However, inference in these models can often be difficult or intractable, motivating a class of variational methods that frame the inference problem as optimization. In particular, Kingma and Welling [2014] propose VAEs to tackle this intractability. Moreover, they also consider a scenario for large datasets: they need a general algorithm that helps to effectively update parameters using small mini-batches. Recall that sampling-based solutions (e.g., Monte Carlo EM) are too slow because they involve a typically expensive sampling loop per data point.

The variational bound: Evidence lower-bound (ELBO) ELBO is the quantity optimized in variational Bayesian methods. These methods handle cases where a distribution over unobserved variables y_1^I is optimized as an approximation to the true posterior $p(y_1^I|e_1^I)$, given observed data e_1^I. ELBO is defined in our case as:

$$
\begin{aligned}
\log p(f_1^J, e_1^I) &= \log \int_{y_1^I} p(e_1^I, f_1^J, y_1^I) dy_1^I \qquad (5.1)\\
&= \log \int_{y_1^I} q(y_1^I) \frac{p(e_1^I, f_1^J, y_1^I)}{q(y_1^I)} dy_1^I \\
&\geq \int_{y_1^I} q(y_1^I) \log[\frac{p(e_1^I, f_1^J, y_1^I)}{q(y_1^I)}] dy_1^I = ELBO
\end{aligned}
$$

Miao et al. [2016] propose a deep neural variational inference framework for generative models of text for document modeling and question-answer selection tasks. Bowman et al. [2016] propose a language model that VAEs help to generate an explicit global distributed sentence representation. In NMT, Zhang et al. [2016] demonstrate translation improvements for long sentences, followed by the work of Pagnoni et al. [2018] which extend VAEs with a co-attention mechanism. The model of Su et al. [2018] introduces a series of latent random variables to model the translation procedure of a sentence in a generative way instead of using just one single latent variable. Eikema and Aziz [2018] introduce a model that generates source and target sentences jointly from a shared latent representation, which is close to our approach.

5.2 Our variants for neural word alignment variational models

Let's recall the standard approach to probabilistic alignment (Section 2.4). This approach is to consider *asymmetric* models associating each word in a source sentence $f_1^J = f_1 \dots f_J$ of J words with exactly one word from the target sentence $e_0^I = e_0 \dots e_I$ of $I+1$ words.[1] This association is governed by unobserved alignment variables $a_1^J = a_1 ... a_J$, yielding the following model:

$$p(f_1^J, a_1^J | e_0^I) = \prod_j^J p(a_j | a_1^{j-1}, f_1^{j-1}, e_0^I) p(f_j | a_1^j, f_1^{j-1}, e_0^I) \tag{5.2}$$

Two versions of this conditional model are considered here: in the IBM model 1 [Brown et al., 1993b], the alignment model $p(a_j | a_1^{j-1}, f_1^{j-1}, e_0^I)$ is uniform; in the HMM model of Vogel et al. [1996], Markovian dependencies between alignment variables are assumed and a_j is independent from all the preceding alignment variables given a_{j-1}. In both models, f_j is conditionally independent to any other variable given a_j and e_1^I. Under these assumptions, both parameter estimation and optimal alignment can be performed efficiently with dynamic programming algorithms. In these conditional approaches, e_1^I is not modeled.

5.2.1 A fully generative model

We present the fully generative approach introduced by Rios et al. [2018]. In this model, the association between a source word f_j and a target word e_i is mediated by a shared latent variable y_i, assumed to represent the joint underlying semantics of mutual translations. In this model, the target sequence e_1^I is also modeled, yielding the following generative story[2] (See Figure 5.1):

1. Generate a sequence y_0^I of d-dimensional random embeddings by sampling independently from some prior distribution e.g. Gaussian: $y_i \sim \mathcal{N}(0, I)$
2. Generate e_1^I conditioned on the latent variable sequence y_1^I: $e_i | y_i \sim \mathrm{Cat}(f(y_i; \theta))$
3. Generate $a_1^J = a_1 ... a_J$ denoting the alignment from f_1^J to y_0^I: uniform distribution $a_j \sim \mathcal{U}(1/I+1)$ or categorical distribution $a_j \sim \mathrm{Cat}(f(a_{j-1}; \theta))$
4. Generate f_1^J conditioned on y_0^I and a_1^J: $f_j | y_0^I, a_j \sim \mathrm{Cat}(f(y_{a_j}; \theta))$

This yields the following decomposition of the joint distribution of f_1^J and e_1^I, where we marginalize over latent variables y_0^I and a_1^J:

$$p(f_1^J, e_1^I) = \int_{y_0^I} p(y_0^I) p_\theta(e_1^I | y_1^I) \Big(\sum_{a_1^J} p_\theta(a_1^J) p_\theta(f_1^J | y_0^I, a_1^J) \Big) dy_0^I \tag{5.3}$$

[1] As is custom, target sentences are completed with a NULL symbol, conventionally at index 0.

[2] We omit the initial step, consisting in sampling the lengths I and J and the dependencies with respect to these variables.

$$\begin{tikzcd} y_0^I \arrow[r] \arrow[dr] & e_1^I \\ a_1^J \arrow[r] & f_1^J \end{tikzcd}$$

Figure 5.1: Generative story: The target sentence e_1^I is generated conditioned on a sequence of random embeddings y_1^I. Generating the source sentence f_1^J requires latent alignments a_1^J.

Directly maximizing the log-likelihood to estimate the parameters is in general intractable, especially when neural networks are used to model the generation of f_1^J and e_1^I. The standard approach in neural generative models [Kingma and Welling, 2014] is to introduce a variational distribution q_ϕ for the latent variables and to optimize the evidence lower-bound (ELBO). Following [Rios et al., 2018], we consider tractable alignment models and use the variational distribution only for modeling y_0^I conditioned on e_1^I. (5.2) yields the following objective:

$$\begin{aligned} J(\theta,\phi) = &- \mathbb{E}_{q_\phi(y_1^I)}(\log\, p_\theta(e_1^I|y_1^I)) - \mathbb{E}_{q_\phi(y_0^I)}([\log \sum_{a_1^J} p_\theta(a_1^J) p_\theta(f_1^J|y_0^I, a_1^J)) \\ &+ \mathrm{KL}[q_\phi(y_0^I|e_1^I)||p(y_0^I)] \end{aligned} \tag{5.4}$$

where $\mathbb{E}_p(f)$ denotes the expectation of f with respect to p, and KL is the Kullback-Leibler divergence. Objective (5.4) is a sum of three terms that are referred to respectively as the *reconstruction cost*, the *alignment cost* and *KL divergence cost*. The last term can be computed analytically when the prior and the variational distributions are Gaussian and we thus assume the following parameterization $q_\phi(y_1^I|e_1^I) = \prod_i N(y_i|u_i, s_i)$, where the mean u_i and the diagonal co-variance matrix diag(s_i) are deterministic functions of e_1^I. As is custom, the expectations in equation (5.4) are approximated by sampling values of y_i as $y_i = u_i + s_i \cdot \epsilon_i$, where ϵ_i is drawn from a white Gaussian noise. The reparameterization trick removes the sampling step from the generation path and makes the whole objective differentiable [Kingma and Welling, 2014].

We clarify here equation (5.4) where y_0^I only conditions on e_1^I:

$$\begin{aligned} ELBO &= \int_{y_0^I} q_\phi(y_0^I) \log[\frac{\sum_{a_1^J} p_\theta(e_1^I, f_1^J, y_0^I, a_1^J)}{q_\phi(y_0^I)}] dy_0^I \qquad (5.5) \\ &= \int_{y_0^I} q_\phi(y_0^I) \log[\frac{p_\theta(y_0^I)}{q_\phi(y_0^I)} p_\theta(e_1^I|y_1^I) \sum_{a_1^J} p_\theta(a_1^J) p_\theta(f_1^J|y_0^I, a_1^J)] dy_0^I \\ &= \int_{y_1^I} q_\phi(y_1^I) \log p_\theta(e_1^I|y_1^I) dy_1^I + \int_{y_0^I} q_\phi(y_0^I) \log[\sum_{a_1^J} p_\theta(a_1^J) p_\theta(f_1^J|y_0^I, a_1^J)] dy_0^I \\ &\quad - \int_{y_0^I} q_\phi(y_0^I) \log[\frac{q_\phi(y_0^I)}{p(y_0^I)}] dy_0^I \\ &= \mathbb{E}_{q_\phi(y_1^I)}(\log\, p_\theta(e_1^I|y_1^I)) + \mathbb{E}_{q_\phi(y_0^I)}([\log \sum_{a_1^J} p_\theta(a_1^J) p_\theta(f_1^J|y_0^I, a_1^J)) \\ &\quad - \mathrm{KL}[q_\phi(y_0^I|e_1^I)||p(y_0^I)] \end{aligned}$$

5.2.2 Introducing Markovian dependencies

The experiments in [Rios et al., 2018] only consider basic assumptions regarding the alignment model $p_\theta(a_1^J)$, corresponding to IBM model 1. Our first variation of this model considers a richer transition model assuming Markovian dependencies, for which the exact marginalization of asymmetrical alignment variables implied by equation (5.4) remains tractable with the forward algorithm. The alignment cost is the expectation of the source given the latent variables:

$$\mathbb{E}_{q_\phi(y_0^J)}([\log \sum_{a_1^J} \prod_{j=1}^{J} p_\theta(f_j|y_{a_j}) p_\theta(a_j|a_{j-1})]) \tag{5.6}$$

As is usual with HMM variants of alignment models, we parameterize the transition distribution $p_\theta(a_j|a_{j-1})$ on the distance (jump) between the values of a_j and a_{j-1} [Och and Ney, 2003]. This model is referred to `HMM+VAE`.

5.2.3 Towards symmetric models: a parameter sharing approach

A first benefit of having a fully generative model (in both alignment directions), which jointly models f_1^J and e_1^I, is that it becomes easy to encourage these models to share information and to improve their joint performance. Our alignment model involves two decoders, one for the source and one for the target (in each direction) (see Figure 5.2). These components are used to compute a distribution over vocabulary words given a d-dimensional variable and are conceptually similar.

Our first step is thus to simultaneously train the alignment models in both directions, making sure that they use the same decoder respectively for f_1^J and e_1^I. This means that the same network computes $p_\theta(e_1^I|y_1^I)$ (when e_1^I is in the target) and $p_\theta(e_1^I|y_0^J, a_1^J)$ when e_1^I is the source. There is only one encoder computing the variational parameters in each direction, and these remain distinct in this approach. Our joint objective function now comprises six terms including two reconstruction costs, two alignment costs and two KL divergence costs. From this, we see that the first benefit of this method is computational as it greatly reduces the number of parameters to train. We also expect that it will yield two additional benefits: (a) to help improve the alignment model, which is more difficult to train for lack of observing the "right" alignment variables; in comparison the reconstruction of the target sentence is almost obvious, as each e_i is generated from the right y_i; (b) to make the alignments more symmetrical, thereby facilitating their interpretation and their recombination. This model is denoted `+VAE+SP` below.

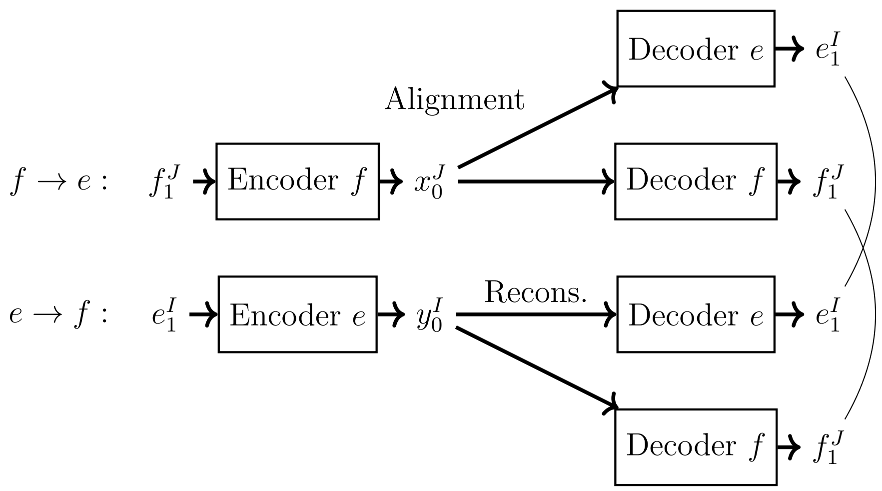


Figure 5.2: Our alignment models involves two decoders, one for the source and one for the target (in each direction). We can simultaneously train the alignment models in both directions, making sure that they use the same decoder respectively for f_1^J and e_1^I.

5.2.4 Enforcing agreement in alignment

The idea of training two asymmetrical models opens new ways to control the level of agreement between alignments, an idea already considered e.g. in [Liang et al., 2006, Graça et al., 2010].

$$\begin{tikzcd} y_0^I \arrow[r] \arrow[dr] & e_1^I & x_0^J \arrow[r] \arrow[dr] & f_1^J \\ a_1^J \arrow[r] & f_1^J & b_1^I \arrow[r] & e_1^I \end{tikzcd}$$

Figure 5.3: Illustration for two asymmetrical models: We enforce agreement between a_1^J and b_1^I

Following the former approach, we implement this idea by adding an extra cost that rewards agreement between asymmetric alignments (see Figure 5.3). For non-null alignment links, this cost is based on the alignment posterior distributions and is defined as:

$$\sum_{i>0,j>0} |p(a_j = i|f_1^J, e_1^I) - p(b_i = j|f_1^J, e_1^I)|, \tag{5.7}$$

where b_1^J is the set of alignment variables introduced when e_1^I is the source of the alignment, and f_1^J is the target. Both for the IBM-1 and the HMM variants, these posterior distributions can be computed effectively, in the latter case using the forward-backward algorithm.

In the case of the null links, the agreement term should reward configurations where one source word is aligned with the null symbol in one direction and is not aligned to any target word in the other direction. This yields the following additional term (for the canonical source to target direction, the reverse term is analogous):

$$\sum_{j=1}^{J} |1 - p(a_j = 0|f_1^J, e_1^I) - \sum_{i=1}^{I} p(b_i = j|f_1^J, e_1^I)| \tag{5.8}$$

For this model (`+VAE+SP+AC`), the objective function comprises nine terms, each with its own dynamics, which makes optimization more difficult due to the heterogeneity between costs.

5.2.5 Training with monolingual data

$$\dot{e}_1^M \rightarrow \boxed{\text{Encoder } e} \rightarrow \dot{y}_1^M \longrightarrow \boxed{\text{Decoder } e} \rightarrow \dot{e}_1^M$$

Figure 5.4: Training with monolingual data through the reconstruction component

Leaving the alignment module aside, the model can be used as a simple autoencoder which can be (pre)trained monolingually (see Figure 5.4). We use supplementary monolingual sentences $\dot{e}_1^M$ that just go through the encoding-decoding process, and add an extra monolingual reconstruction term J_{mono} in the objective (5.4):

$$J_{\text{mono}}(\theta, \phi) = -\mathbb{E}_{q_\phi(\dot{y}_1^M)}(\log p_\theta(\dot{e}_1^M|\dot{y}_1^M)) + \text{KL}[q_\phi(\dot{y}_1^M|\dot{e}_1^M)||p(\dot{y}_1^M)] \tag{5.9}$$

where $\dot{y}_1^M$ is the latent variable associated with $\dot{e}_1^M$. Alternatively, we consider training the alignment model monolingually. We implement this idea by adding a random noise to the target sentence, to make it more similar to a source sentence and amenable to alignment. In this case, the extra reconstruction term is:

$$J_{\text{mono}}(\theta, \phi) = -\mathbb{E}_{q_\phi(\ddot{y}_0^N)}([\log \sum_{\ddot{a}_1^M} p_\theta(\ddot{a}_1^M) p_\theta(\dot{e}_1^M|\ddot{y}_0^N, \ddot{a}_1^M)) + \text{KL}[q_\phi(\ddot{y}_0^N|\ddot{e}_1^N)||p(\ddot{y}_0^N)] \tag{5.10}$$

where $\ddot{e}_1^N$ is a noisy version of $\dot{e}_1^M$, $\ddot{y}_1^N$ is the latent variable for $\ddot{e}_1^N$. $\ddot{a}_1^M$ denotes the alignment variables between $\dot{e}_1^M$ and $\ddot{y}_0^N$. Note that these alignment variables $\ddot{a}_1^M = (\ddot{a}_1, \ldots, \ddot{a}_M)$ with $\ddot{a}_m \in [0 \ldots N]$ help to reproduce the original sentence $\dot{e}_1^M$ from its noised sentence $\ddot{e}_1^N$. In our experiments, we only use IBM Model 1 as our alignment model: $\ddot{a}_m \sim \mathcal{U}(1/N + 1)$.

5.3 Experiments

For our variational models, we perform the alignment between subword units. This helps to eliminate unknown words and reduce the problem of rare words. Moreover, we get rid of the complex architecture of our above-mentioned model `NN+CharBoth` where pure character-based representations on both sides are considered (Chapter 4). This is also an initial step to explore subword alignments that we latter discuss in Chapter 6.

Following notably [Garg et al., 2019], we perform the alignment between subword units generated by Byte-Pair-Encoding [Sennrich et al., 2015], implemented with the SentencePiece model [Kudo and Richardson, 2018] and computed independently in each language with 32K merge operations[3]. For Vietnamese, we use 16K merge operations[4]. This makes the training less computationally demanding and greatly mitigates the rare-word problem, which is a major weakness of historical count-based models. Our results and analyses are however based on word-level alignments. Subword-level alignments are converted into word-level alignments as follows: a link between a source and a target word exists if there is at least one link alignment between their subwords (Section 2.3.2). In all cases, our optimizer is Adam [Kingma and Ba, 2014] with an initial learning rate of 0.001; the batch size is set to 100 sentences. We use all training sentences of length lower than 50. We train all models for 10 iterations. Results with symmetric alignments use the grow-diag-final (GDF) heuristic proposed in [Koehn et al., 2005].

In our experiments, we use Python version 3.6, Numpy version 1.2 and Tensorflow version 1.0.1. The implementation is available from `https://github.com/ngohoanhkhoa/Generative_Probabilistic_Alignment_Models`.

Architecture Our models are close in structure to the model proposed by Rios et al. [2018], and are made of three main components: an encoder to generate the latent variables y_0^I from e_1^I, and two decoders to respectively reconstruct e_1^I and f_1^J, with the help of the alignment model. The architecture of this fully generative model is displayed in Figure 5.5.

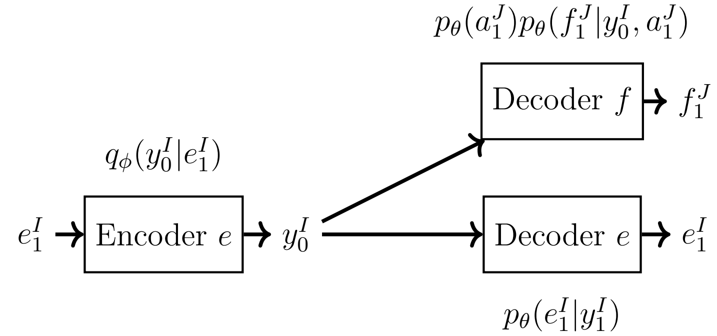


Figure 5.5: Architecture of a fully generative model: an encoder to generate the latent variables y_0^I from e_1^I, and two decoders to respectively reconstruct e_1^I and f_1^J, with the help of the alignment model.

- The encoder is composed of a token embedding layer (128 units), two LSTM layers (each comprising 64 units), and dense output layers to independently generate the mean vectors $(u_1 \ldots u_I)$ vectors and the diagonal of the covariance matrices $(s_1 \ldots s_I)$. The latent variable y_1^I has 64 units. Our encoder is formally defined as:

$$\overrightarrow{h_i} = RNN(\overrightarrow{h_{i-1}}, E(e_i)) \qquad s_i = \text{softplus}(W_s h_i + b_s)$$
$$h_i = W_h \,\text{concat}(\overrightarrow{h_i}, \overleftarrow{h_i}) \qquad u_i = W_u h_i + b_u$$
$$y_i = u_i + s_i \cdot \epsilon_i$$

[3]We differ there from Garg et al. [2019] who use a joint BPE vocabulary.
[4]The vocabulary size of Vietnamese training corpus is ∼19K words (Table 3.2)

where $E(e_i) \in \mathbb{R}^{128}$ is the embedding of word e_i, ϵ is a noise variable $\epsilon \sim N(0,1)$ and softplus $= \log(1+\exp(x))$ is an activation function returning a positive value (Section 4.1). The vector y_0 is independently generated from a pseudo-sentence made of one dummy token; it is identical for all target sentences. Note that the decoder model does not try to reconstruct this token.

- The reconstruction decoder is given by:

$$p_\theta(e_i|y_i) = [\text{softmax}(W_v y_i + b_v)]_{e_i}$$

and the alignment model with emission and transition components is:

$$p_\theta(f_j|e_{a_j}) = [\text{softmax}(W_v y_{a_j})]_{f_j}$$
$$p_\theta(a_j - a_{j-1}) = [\text{softmax}(W_\Delta y_{a_{j-1}})]_{a_j - a_{j-1}}$$

where $W_v \in \mathbb{R}^{64\times V}$, $b_v \in \mathbb{R}^V$, with V the target vocabulary size. $W_\Delta \in \mathbb{R}^{64\times 301}$ with jump values in the interval $[-150, +150]$.

Baselines All parameters of the `Giza++` and `Fastalign` baselines are set to their default values. `IBM-1+NN` and `HMM+NN` correspond to basic neuralizations of the IBM/HMM models as in Section 4.3 for word-level, character-level and BPE-level. Note that `+B` uses an architecture similar to `+VAE`: Its neural translation/distortion model is based on an architecture composed of a token embedding layer (128 units), two LSTM layers (each comprising 64 units), a dense layer, followed by a drop-out layer and a softmax layer. These models are trained by maximizing the likelihood with the expectation-maximization algorithm.

Noise model For experiments with monolingual data, our noise model follows the technique of Lample et al. [2017]. We randomly delete input words with probability $p_{wd} = 0.1$. We then slightly shuffle the sentence, where the difference between the position before and after shuffling each word is smaller than 4. Figure 5.6 displays an example of adding noise into target sentences.

Source:	it	is	quite	understandable	.
Step 1:	it		quite	understandable	
Step 2:	it		understandable	quite	

Figure 5.6: Example for the noise model proposed in [Lample et al., 2017]: (Step 1) Randomly delete input words with probability $p_{wd} = 0.1$, (Step 2) Slightly shuffle the sentence, where the difference between the position before and after shuffling each word is smaller than 4.

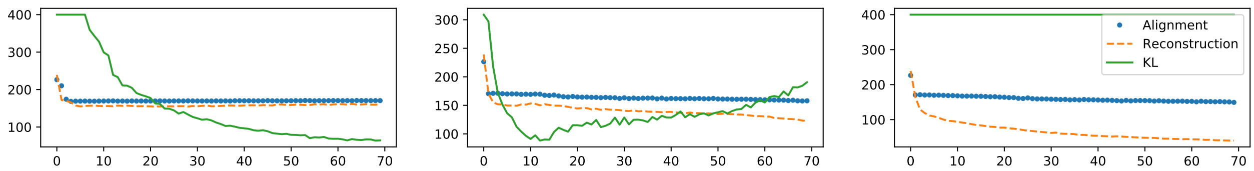
Balancing the terms in the VAE objective One well-known issue of VAEs for text applications is *posterior collapse* [Bowman et al., 2016, Higgins et al., 2017], where the variational distribution collapses towards the prior distribution.

This is because the KL term can get arbitrarily small, with a moderate effect on the reconstruction cost, assuming a strong reconstruction model (a recurrent network in typical applications). We also encountered this problem in our setting, but the interpretation is a bit different: when the KL term goes to zero, all words in the dictionary become indistinguishable and the reconstruction cost reaches its maximum, corresponding to the entropy of the uniform distribution of the target vocabulary. The difference in dynamics between these scores is observed in Figure 5.7 (left), where we apply weights equal to α, β and γ respectively to the reconstruction cost, the alignment cost and the KL divergence term. This effect is mitigated if we proportionally decrease the weight of the KL term (middle). This second graph reveals

α	β	γ	AER
1	1	1	92.23
0.6	0.3	0.1	92.35
0.8	0.15	0.05	67.16
2	2	0.5	75.04
20	10	0.5	60.83
50	10	0.5	55.15
100	10	0.5	55.10
10	**50**	0.5	53.22
10	100	0.5	53.31

Table 5.1: Searching for the right balance of weights in the objective function

the need to also better balance the importance of the other two terms. Using larger weights for the reconstruction term ($\alpha = 10$) and even more for the alignment term ($\beta = 50$), we keep the KL divergence high and make sure that the optimization focuses on decreasing the two other terms. In our baseline experiment with the development corpus (English-Romanian), using these weights resulted in acceptable AER scores and seemed appropriate for our further experiment. A small exploration of the hyper-parameter space showed that these results were stable (see Table 5.1).



$\alpha = 1, \beta = 1, \gamma = 1$ $\alpha = 1, \beta = 1, \gamma = 0.5$ $\alpha = 10, \beta = 50, \gamma = 0.5$

Figure 5.7: Visualizing the three terms of the ELBO for Romanian-English. The weights of the reconstruction cost, alignment cost and KL divergence are set to α, β, γ respectively.

5.4 Evaluation

In this section, we perform a detailed analysis of the quantitative results discussed in Chapter 3, focusing mostly on the benefits of variational versions `HMM+VAE` and `IBM+VAE` models, operating at the BPE level. We also report the performance of the baselines: the count-based model (`Giza++`) and the several neural variants (Section 4.3), operating at the word (`+NN`), subword (`+BPE`) and character levels. Our goal in this section is to better understand the improvements brought by this kind of model, but also to identify the weaknesses of variational models for the task of word alignment. Note that we only show the results of our models that greatly differ from the results of their counterparts. Complete results are in [Ngo Ho, 2021, Appendix D] (Reporting AER scores as a function of sentence length and sentence length difference, or sorted by main syntactic tags are displayed in [Ngo Ho, 2021, Appendix D.11], [Ngo Ho, 2021, Appendix D.12] and [Ngo Ho, 2021, Appendix D.8]).

5.4.1 AER, F-score, precision and recall

Table 5.2 reports the AER score of the `IBM-1` baselines, several variants of `IBM-1+NN` and our variational models. A first observation is that neural baselines are better than Giza++, and that using BPE units brings an additional gain. The basic model (`IBM-1+VAE`) falls short to match these results and proves way worse than the neural version of the IBM-1 model. These results are in line with the findings of Rios et al. [2018], who report similar differences in

Corpus	`Giza++`	`+NN`	`+NNChar`	`+NN+BPE+B`	`+BPE+VAEs`		
					Vanilla	+SP	+SP+AC
English-French	40.1	27.96	28.76	25.71	*33.42*	***22.12***	22.87
French-English	33.9	27.21	31.4	24.05	34.36	23.89	**23.61**
English-German	39.03	37.64	36.22	31.36	*38.92*	*24.41*	**24.3**
German-English	42.66	39.22	40.88	34.46	40.87	38.72	**29.37**
English-Romanian	56.02	46.4	50.16	**43.47**	*56.39*	*49.3*	49.12
Romanian-English	53.52	44.9	48.28	**40.42**	55.7	51.49	49.2
English-Czech	45.09	42.29	40.85	**30.76**	*41.92*	*39.61*	35.41
Czech-English	48.47	40.97	42.35	**32.71**	45.3	42.63	33.83
English-Japanese	63.12	62.64	57.96	56.51	58.66	55.54	**54.81**
Japanese-English	61.55	56.9	54.91	57.27	*59.95*	*55.1*	**54.23**
English-Vietnamese	69.43	58.87	55.06	55.85	56.47	51.34	**50.84**
Vietnamese-English	46.45	42.25	41.15	**37.72**	*53.56*	*41.38*	38.77

Table 5.2: AER score of our VAE models compared with the corresponding `IBM-1` baselines.

performance. Sharing the parameters between directions greatly improves this baseline with a reduction in AER (about 11 points for English-French, about 14/2 points for English-German, about 7/4 points for English-Romanian, about 2 points English-Czech, about 3/4 points for English-Vietnamese, about 5/12 points for English-Japanese in both directions).

The reconstruction model, which is well trained in one direction, helps to improve the emission model in the reverse direction. We observe that the gain is more significant when the morphologically rich language (e.g., French, German, Romanian, Czech) is on the target side: this is where the emission model is the weakest and benefits most from parameter sharing. For Japanese, we see the opposite effect. This can be because English is morphologically richer than Japanese. In the case of English-Vietnamese, the reconstruction model for English proves very useful, leading the best score of 50.84 AER.

Adding an extra agreement cost helps to produce markedly better alignments except for English-French. Moreover, this approach brings larger gains when English is in the target side. Its best AER scores can be found in English-German and English-Japanese on both sides. Overall, our best VAE model outperforms the neural baseline `+NN+BPE+B` in a large training condition (i.e., English-French and English-German). We do not see this for the other language pairs with the small training condition (except for English-Japanese), where the performance remains much below the neural baseline.

Corpus	`Fastalign`	`Giza++`		`+NN`	`+NNCharJB`	`+NN+BPE+B`	`+BPE+VAEs`		
		HMM	IBM-4				Vanilla	+SP	+SP+AC
English-French	15.19	11.99	10	11.84	**8.47**	9.84	18.92	12.94	11.47
French-English	16.23	11.97	9.64	11.15	**7.74**	10.48	12.94	12.27	10.84
English-German	28.98	23.92	21.46	26.78	23.69	19.61	23.96	23.73	**19.13**
German-English	31.28	26.33	23.31	29.44	24.9	**20.38**	26.5	26.4	20.58
English-Romanian	33.36	33.36	31.04	30.69	**26.85**	34.41	50.29	37.52	35.55
Romanian-English	32.91	36.38	32.3	40.12	29.76	**29.34**	38.64	38.04	38.87
English-Czech	25.75	27.86	20.92	23.5	16.38	**16.24**	23.71	20.31	17.56
Czech-English	25.3	30.38	26.5	24.06	24.61	**18.74**	29.01	20.12	20.1
English-Japanese	50.67	57.01	52.52	49.68	40.92	**38.33**	49.27	43.67	40.86
Japanese-English	49.37	54.41	49.23	47.09	**37.71**	38.93	53.78	48.99	45.24
English-Vietnamese	48.89	57.86	51.91	49.27	**43.28**	47.03	48.97	45.87	43.94
Vietnamese-English	32.82	37.57	33.19	31.45	**27.59**	27.76	39.2	33.78	32.59

Table 5.3: AER score of our VAE models compared with the corresponding `HMM` baselines.

We observe the effect of adding a transition component in Table 5.3. Our variational models outperform their discrete counterparts in most cases (almost -10 AER). Both symmetrization strategies prove again very effective to improve the basic VAE model, and our best system (`+AC`) achieves AER scores that are close, yet slightly inferior, to the `HMM+NN+BPE+B` and `HMM++NNCharJB` baseline. Note that it yields the best result in the case of English-German. One possible issue that we do not fully solve via symmetrization is related to the null word, which, as explained above, is not part of the reconstruction model, and which does not improve with joint learning.

5.4.2 Are unaligned words still a problem ?

In asymmetrical models, the number of links that are generated is constant and equal to the total number of "source" words. A source word is deemed unaligned when it is linked to the special NULL token on the target side; a target word is unaligned when it emits no source word. We perform an in-depth analysis of these special links. Results for the alignment from French into English are in Figure 5.8; we observe similar trends for other language pairs and for both directions. We see that the number of unaligned words (on both sides) varies in great proportion, with a minimum of about 3600 words (`IBM-1+BPE+B`) and a maximum of nearly 6000 (`IBM1+BPE+VAE` and `HMM+VAE+BPE`). For this language pair, the reference contains 821 unaligned words. They also demonstrate the inability of all models to correctly predict null links, the best model achieving a precision of only about 13%.

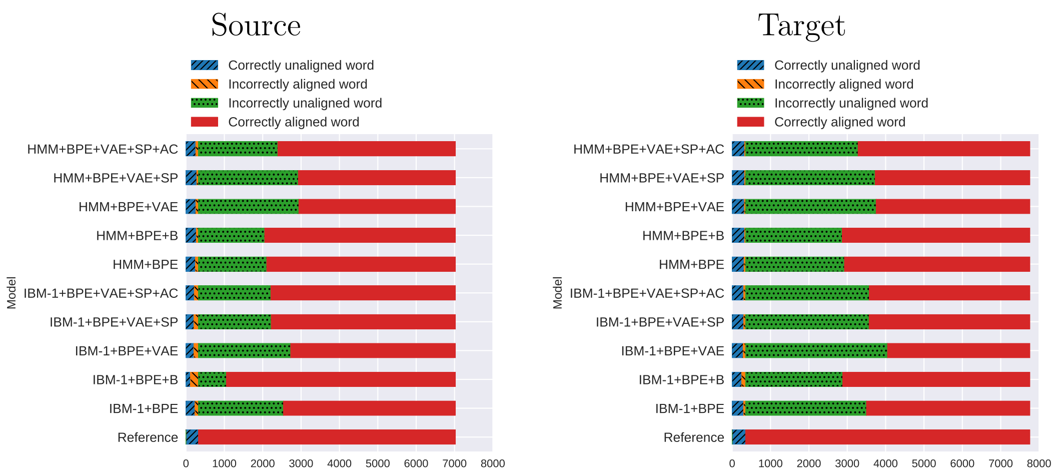


Figure 5.8: Results of our variational models: Unaligned words for the direction English-French

Predicting so many unaligned words is extremely detrimental to the performance of the two basic VAE models for which we observe a very poor recall for non-null links, which is hardly compensated by the good precision scores. We see here clearly the effect of the symmetrization constraints (especially for the HMM model) where the reward associated with symmetric predictions reduces the tendency to align French words with the NULL English and to leave too many English words unaligned. Even there (`HMM+VAE+BPE+SP+AC`), the number of predicted non-null links is about half as what we see for `HMM+NN`: as it predicts much more links than the others, this model also as a clear edge when it comes to post-hoc symmetrization since the "grow-diag-final" heuristics heavily depends on the size of the intersection. Note that this problem has a much stronger overall effect in language pairs whose test sets only contain sure links than English-French. In order words, a low recall for aligned words directly impacts the AER. We do not see this for the French-English data, which contains many possible links that have no impact on recall [Fraser and Marcu, 2007].

Incidentally, we also observe a null-word problem for `HMM+NN+BPE` (`HMM+BPE` in Figure 5.8); presumably splitting words into small units that are unrelated across languages can also make the model prefer the null alignment over links between actual words. These results clearly point out one deficiency of the current approach: for lack of having a proper model for the latent representation of the NULL token, the VAE-based approach tends to leave too many words unaligned.

5.4.3 Symmetrization and agreement

We now study the effects of sharing parameters across alignment directions. We consider the English-Romanian test, for which the relationship between precision, recall, and AER is straightforward. Detailed scores for all variational models and several baselines are in Table 5.4. We see the clear benefits of sharing parameters, which contribute a jump of both precision, recall, and F-measure compared with the baseline VAE. Models `SP` and `SP+AC` generate more alignment links (about +500 links) than the baseline model. This enhancement helps to outperform `Giza++` but is insufficient to surpass the conventional neural network models, especially when using BPE. Numbers in Table 5.4 show that the gain in recall is largest in the direction English-Romanian: this is because the better reconstruction of English words boosts the translation model.

Models	English-Foreign			Foreign-English			GDF		
	F1	PRE	REC	F1	PRE	REC	F1	PRE	REC
IBM-1 Giza++	43.99	58.8	35.14	46.49	49.92	43.5	48.88	73.82	36.54
IBM-1+NN	53.62	57.71	50.07	55.11	60.08	50.9	61.64	75.8	51.94
IBM-1+NNChar	49.85	54.28	46.09	51.73	56.08	48.01	58.6	75.25	47.98
IBM-1+BPE	56.25	**79.61**	43.49	56.05	**70.8**	46.39	58.06	78.17	46.18
IBM-1+BPE+B	**56.54**	63.95	**50.67**	**59.59**	64.19	**55.61**	**65.56**	**80.47**	**55.31**
IBM-1+BPE+VAE	43.63	56.66	*35.47*	44.32	53.94	*37.61*	48.67	79.6	35.05
IBM-1+BPE+VAE+SP	50.71	60.69	*43.55*	48.52	57.82	*41.8*	54.81	76.23	42.79
IBM-1+BPE+VAE+SP+AC	50.89	61.31	*43.5*	50.81	59	*44.62*	56.65	76.91	44.84
Fastalign	66.65	72.77	61.49	67.1	73.7	61.59	69.6	72.65	66.8
HMM Giza++	66.65	75.28	59.8	63.64	72.9	56.46	67.62	76.63	60.5
HMM+NN	69.33	76.93	63.09	59.89	63.85	56.4	65.66	65.89	65.43
HMM+NNCharTgt	72.47	78.13	**67.59**	**72.01**	82.79	**63.71**	**74.05**	79.04	**69.66**
HMM+NNCharJB	**73.17**	83.55	65.08	70.26	80.7	62.21	73.89	83.22	66.43
HMM+BPE	65.79	**84.07**	54.04	69.27	82.44	59.74	69.76	83.34	59.99
HMM+BPE+B	65.61	84.04	53.81	70.68	82.91	61.59	70.57	83.31	61.21
HMM+BPE+VAE	49.73	75.24	37.14	61.38	79.8	49.87	57.29	83.13	43.7
HMM+BPE+VAE+SP	62.5	87.99	48.46	61.98	**88.3**	47.75	62.99	**91.62**	48
HMM+BPE+VAE+SP+AC	64.47	81.66	53.26	61.15	78.09	50.25	64.83	84.51	52.59
IBM-4 Giza++	68.98	79.28	61.04	67.72	80.97	58.2	70.94	82.98	61.96

Table 5.4: Grow-diag-final: F-score (F1), precision and recall (%) for English-Romanian

We now measure more directly the level of agreement between the two alignment directions for English-French (Table 5.5). We note that the model integrating agreement costs (`+SP+AC`) leads to a higher number of agreements in comparison to the other VAE-based models, and also yields the best scores in terms of intersection AER. Complete results are in [Ngo Ho, 2021, Appendix D.9].

Models	# links	Ratio		AER	F1	PRE	REC	ACC	FE	
		En-XX	XX-En						En	Fr
Fastalign	4879	0.69	0.75	11.09	40.48	92.58	25.9	90.71	0.7	0.73
HMM Giza++	4683	0.73	0.76	7.59	41.16	97.2	26.1	90.9	0.65	0.82
HMM+NN	4771	0.73	0.77	7.42	41.53	96.67	26.45	90.92	0.57	0.64
HMM+NNCharTgt	5049	0.78	0.81	6	43.54	96.95	28.07	91.12	0.43	0.64
HMM+NNCharJB	4698	0.8	0.85	6.27	41.62	98.06	26.42	90.96	0.39	0.64
HMM+BPE	3898	0.72	0.78	11.54	36.09	98.77	22.08	90.46	0.65	0.64
HMM+BPE+B	4040	0.75	0.8	10.5	37.12	98.66	22.86	90.56	0.65	0.64
HMM+BPE+VAE	3160	0.69	0.76	18.73	30.16	98.29	17.81	89.94	0.48	0.64
HMM+BPE+VAE+SP	3586	0.86	0.87	13.09	33.5	98.22	20.2	90.22	0.61	0.55
HMM+BPE+VAE+SP+AC	3989	0.84	0.85	10.17	36.35	97.62	22.33	90.46	0.65	0.55
IBM-4 Giza++	4588	0.77	0.81	7.76	40.88	98.13	25.82	90.89	0.65	0.64

Table 5.5: Intersection alignment for variational models: The number of alignment links, their ratio to the total number of alignment links predicted by the model, alignment error rate (AER), accuracy (ACC), F-score (F1), precision (PRE), recall (REC) and average fertility (FE) for English-French.

5.4.4 Training with monolingual data

The last extension concerns the use of monolingual data in the low-resource condition. Experiments are performed with English-Romanian: the Romanian corpus is from News Crawl 2019 (∼6.7M sentences) and the English corpus is from Europarl, and corresponds to the English side of the English-French data.

Results are in Table 5.6. Note that to compute the performance of the reconstruction model (R-ACC), we compute the proportion of words for which the model's prediction actually corresponds to the correct word. We see that `+Mono` helps improve the reconstruction model, which attains almost perfect reconstruction accuracy in both directions, suggesting that the auto-encoder is over-fitting. The gain brought by monolingual data is found only for `IBM-1`, for the direction Ro-En (-3.6 AER). The extra-task of denoising the input (`+Mono+Noise`) further improves the AER compared to the parameter sharing approach.

Models	English-Foreign					Foreign-English				
	AER	F1	PRE	REC	R-ACC	AER	F1	PRE	REC	R-ACC
	IBM-1+BPE+VAE									
+SP	49.3	50.71	60.69	43.55	84.6	51.49	48.52	57.82	41.8	93.0
+Mono	49.1	50.91	59.3	44.61	98.1	47.89	52.03	61.21	45.24	96.43
+Noise	48.9	51.11	59.9	44.57	98.4	47.63	52.39	59.97	46.51	96.85
	HMM+BPE+VAE+SP									
+SP	37.52	62.5	87.99	48.46	95.5	38.04	61.98	88.3	47.75	97.5
+Mono	37.96	62.05	69.2	56.25	95.5	38.02	61.99	65.66	58.72	97.31
+Noise	36.93	63.08	71.49	56.45	98.8	36.49	63.53	68.39	59.32	97.4

Table 5.6: Training with a monolingual corpus (`+Mono`) and the noise model (`+Noise`) on English-Romanian corpus. R-Acc is the accuracy of the reconstruction model.

We also report the performance of this model `+Mono+Noise` for the English-Czech, the English-Japanese and the English-Vietnamese language pair in [Ngo Ho, 2021, Appendix D.10]. Note that we use the same English corpus in the English-Romanian experiment for these experiments. The Czech monolingual corpus is from Europarl (∼597K sentences). For English-Japanese and English-Vietnamese, we only train the reconstruction component of English side.

We see a gain of about -1/2 AER point for English-Czech and English-Japanese (in both directions) and Vietnamese-English. Table 5.7 displays results for English-Czech. The largest gain (about -3 AER) is also found in the case of `IBM-1` for the direction Czech-English. We can gain some more AER points without a large increase of reconstruction accuracy. This underlines the benefit of the noise model. In the direction English-Vietnamese, we do not see an improvement for `IBM-1+BPE+VAE+Mono+Noise`, which suggest a necessary of a monolingual corpus for Vietnamese.

Models	English-Foreign					Foreign-English				
	AER	F1	PRE	REC	R-ACC	AER	F1	PRE	REC	R-ACC
	`IBM-1+BPE+VAE`									
`+SP`	39.61	49.2	61.7	40.91	61.90	42.63	47.6	55.41	41.72	76.28
`+Mono+Noise`	37.25	53.6	61.22	47.67	62.04	39.28	51.12	58.84	45.19	76.43
	`HMM+BPE+VAE+SP`									
`+SP`	20.31	69.01	82.62	59.25	97.05	20.12	67.94	84.46	56.83	97.22
`+Mono+Noise`	19.11	69.25	86.8	57.6	97.39	18.51	68.64	88.95	55.88	97.28

Table 5.7: Training with a monolingual corpus and the noise model (`+Noise`) on English-Czech corpus. R-Acc is the accuracy of the reconstruction model.

5.4.5 Do symmetrization heuristics improve distortion ?

Figure 5.9 shows jump errors generated by `HMM+BPE+B, HMM+BPE+VAE, HMM+BPE+VAE+SP`. Most jumps of length 0 and jumps to NULL are incorrect in BPE-level. An explanation for jumps of length 0 can be that our distortion model does not recognize boundaries between words and the word recombination process creates a large number of incorrect jumps equal to 0. As mentioned in Section 5.4.2, models `+VAE` have a marked tendency to generate NULL words, and accordingly to jump to "NULL" states, which weakens the performance of these models. We see that sharing parameters does help to reduce the number of incorrect jumps to NULL and jumps of length 1. Especially, adding an agreement cost not only greatly reduce the number of incorrect jumps equal to 0, but yields a large increase of +500 correct jumps of length 1. This suggests that the agreement between two asymmetrical alignments significantly improves short jumps. Similar observations are also found for other language pairs and in both directions [Ngo Ho, 2021, Appendix D.5].

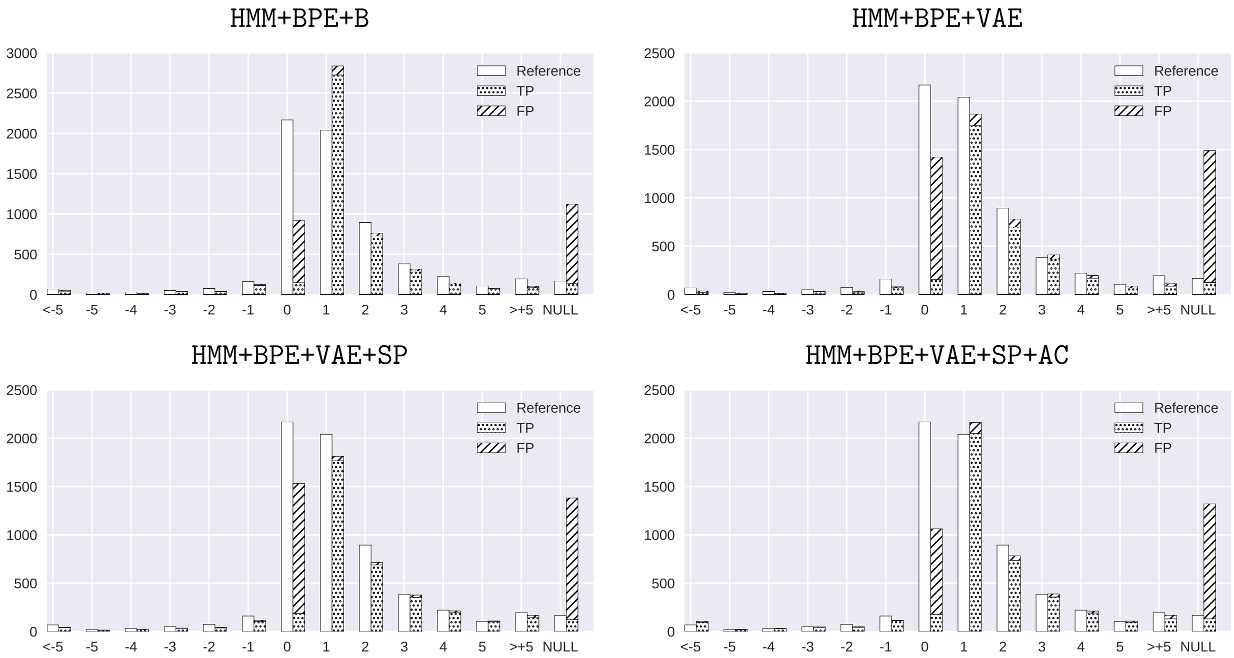


Figure 5.9: Models for the direction English-French: Correct (TP) and incorrect (FP) jump widths for source words on the left graph.

5.4.6 Many-to-many links in BPE-based variational models

We study how BPE-based variational models affect many-to-many links and one-to-many links. As can be seen in Figure 5.10, the BPE-based models generate a very small number of many-to-many/one-to-many links. As mentioned in Section 4.7.5, all `HMM` models encourage one-to-one alignments which accounts for most of the correct links. Using a BPE tokenization and a post-processing step to transform from BPE alignments to word alignments, do not help to create more many-to-many links or one-to-many links. This suggests that we need to find better methods for recombining alignment links between BPE units.

We see two opposite trends[5]

- European languages: There are more one-to-many/many-to-many links when English is on the target side. This is the case of German-English, Romanian-English and Czech-English (Figure 5.10). This behavior can be explained: decomposing a source word of a morphologically richer language (i.e., richer than English) clearly helps an asymmetrical alignment model to generate more of these links. Recall that the vocabulary sizes of these languages are much larger than the corresponding English (see Table 3.2).

  This trend is less clear for English-French where the number of these links is much smaller. An explanation is that English and French are morphologically close and the difference between their vocabulary sizes is small (see Table 3.2).

- Asian languages: The opposite trend is found in Vietnamese and Japanese (Figure 5.11). This is clearly because English is morphologically richer than Japanese and Vietnamese is an isolating language that has no inflectional morphology.

[5] Complete results are found in [Ngo Ho, 2021, Appendix D.4]

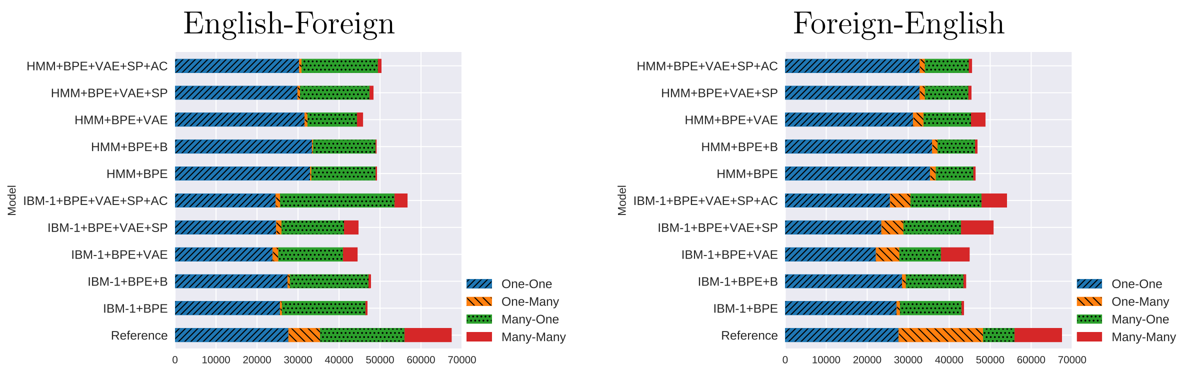


Figure 5.10: Results of our variational models: Alignment types of English-Czech

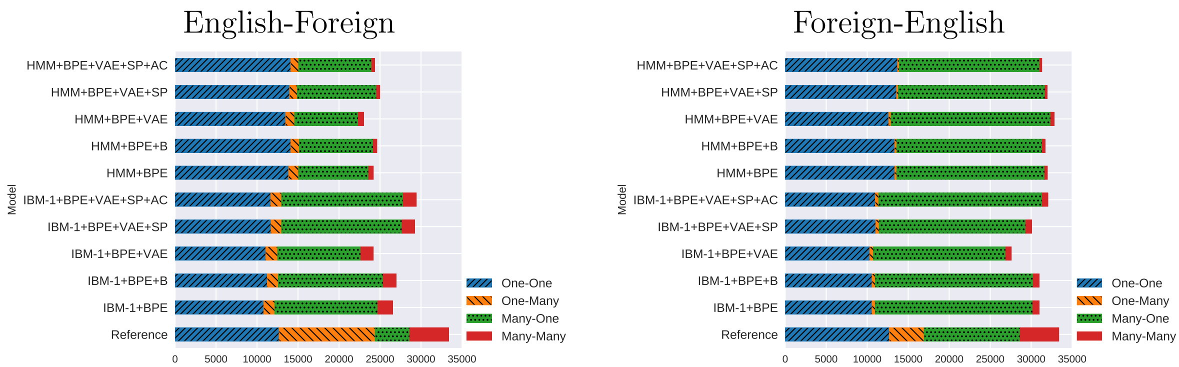


Figure 5.11: Results of our variational models: Alignment types of English-Japanese

5.4.7 Rare/unknown words in BPE-based variational models

We explore how subwords help to get rid of rare/unknown words (Section 3.6 and Section 3.7). For the discrete baselines, we report the performance that we concatenate training and test corpus, which means there is no unknown word. Complete results for rare words and unknown words are respectively in [Ngo Ho, 2021, Appendix D.6] and [Ngo Ho, 2021, Appendix D.7] respectively.

An observation is that using BPE-level alignments and `+VAE` greatly improve the performance (F-score) compared with the discrete and neural baselines. Although a loss in recall, the BPE-based models obtain a large gain in precision, yielding a better F-score. In Table 5.8, we observe unknown words for the English-French language pair where we see the smallest improvement.

For the variants of `IBM-1`, the BPE-based model without Bi-LSTM `IBM-1+BPE` obtains better F-scores than their word-based/character-based counterparts (about +20 points for F-score). Note that using Bi-LSTM does not helps to greatly improve the performance for unknown words. We also see a gain of about 6 points for our variants `+SP` and `+AC` compared with the vanilla variational model `IBM-1+BPE+VAE`. The improvement is less clear for the variant of `HMM` in the direction French-English because of a large loss in recall. This again highlight the problem of our NULL model. Similar behavior can be found in other language pairs/for both directions for rare and unknown words.

Models	English						Foreign					
	#	FE	ACC	PRE	REC	F1	#	FE	ACC	PRE	REC	F1
`IBM-1 Giza++`	680	4.33	79.06	22.65	52.38	31.62	298	4.66	78.04	21.14	62.38	31.58
`IBM-1+NN`	128	0.82	89.37	32.81	14.29	19.91	93	1.45	90.02	37.63	34.65	36.08
`IBM-1+NNChar`	188	1.2	89.31	37.77	24.15	29.46	109	1.7	88.42	30.28	32.67	31.43
`IBM-1+BPE`	166	1.06	91.95	61.45	34.69	44.35	69	1.08	94.53	73.91	50.5	60
`IBM-1+BPE+B`	189	1.2	91.48	56.08	36.05	43.89	76	1.19	93.32	61.84	46.53	53.11
`IBM-1+BPE+VAE`	184	1.17	91.64	57.61	36.05	44.35	100	1.56	91.87	50	49.5	49.75
`IBM-1+BPE+VAE+SP`	183	1.17	92.49	65.03	40.48	49.9	88	1.38	93	57.95	50.5	53.97
`IBM-1+BPE+VAE+SP+AC`	190	1.21	92.58	65.26	42.18	51.24	91	1.42	92.76	56.04	50.5	53.12
`Fastalign`	269	1.71	91.86	56.51	51.7	54	82	1.28	93.81	64.63	52.48	57.92
`HMM Giza++`	432	2.75	88.05	40.05	58.84	47.66	226	3.53	83.67	27.43	61.39	37.92
`HMM+NN`	176	1.12	92.46	65.34	39.12	48.94	94	1.47	93	57.45	53.47	55.38
`HMM+NNCharTgt`	153	0.97	93.43	77.78	40.48	53.24	70	1.09	94.45	72.86	50.5	59.65
`HMM+NNCharJB`	128	0.82	93.59	85.16	37.07	51.66	71	1.11	93.56	64.79	45.54	53.49
`HMM+BPE`	152	0.97	92.64	69.74	36.05	47.53	66	1.03	94.77	77.27	50.5	61.08
`HMM+BPE+B`	144	0.92	93.27	77.78	38.1	51.14	62	0.97	94.93	80.65	49.5	61.35
`HMM+BPE+VAE`	188	1.2	91.83	59.04	37.76	46.06	64	1	93.81	68.75	43.56	53.33
`HMM+BPE+VAE+SP`	138	0.88	93.65	83.33	39.12	53.24	63	0.98	95.33	84.13	52.48	64.63
`HMM+BPE+VAE+SP+AC`	178	1.13	93.02	70.22	42.52	52.97	70	1.09	96.06	87.14	60.4	71.35
`IBM-4 Giza++`	388	2.47	88.81	42.01	55.44	47.8	194	3.03	86.4	32.47	62.38	42.71

Table 5.8: Models for English-French: # links, fertility (FE), accuracy (ACC), F-score (F1), precision (PRE) and recall (REC) for the unknown target words in the direction French-English and in English-French.

5.5 Summary

In this chapter, we revisited the proposal of Rios et al. [2018] and explored variants of the variational autoencoder models for the unsupervised estimation of neural word alignment models. Our study [Ngo Ho and Yvon, 2020] confirmed the previous findings and highlighted two promising aspects of this model:

- It is a full model of the joint distribution, which makes it easy and natural to introduce symmetrization constraints, as shown by our two proposed extensions. With these constraints, we were experimentally able to close the gap with strong baselines implementing neural variants of the conditional HMM models in a large data condition.
  - We encouraged the two asymmetrical models to share information and to improve their joint performance by sharing parameters of the two decoders, one for the source and one for the target in each direction (Section 5.2.3). Note that these decoders are used to compute a distribution over vocabulary words given a d-dimensional variable, and are conceptually similar. We see the improvements in the emission model in one direction thanks to the reconstruction model which is well trained in the reverse direction. The gain is more significant when the morphologically rich language is on the target side where the emission model is the weakest and benefits most from parameter sharing.
  - Based on an idea already considered e.g. in [Liang et al., 2006, Graça et al., 2010], we implement agreement by adding the two extra costs that reward agreement between asymmetric alignments (Section 5.2.4). We observe that this yields a higher level of agreement in comparison to the other VAE-based models and also yields better scores in terms of intersection AER.
- It opens new alleys to also incorporate monolingual data during training, which might especially prove useful in low-resource scenarios.

In addition, we summarize some of our findings based on our evaluation tools:

- One problem of this variational approach is the prediction of null links, which is quite difficult in an encoder-decoder approach. We showed in particular that the VAE model is strongly inclined to under-generate alignment links, which is detrimental to the overall AER performance. Symmetrization is a first answer to this problem, which however only partly fixes the issue. We suggested that we still need a proper model for the latent representation of the NULL token.

- Using BPE-based alignment did not help to create a large number of one-to-many or many-to-many links. An explanation is that splitting words into small units that are unrelated across languages can also make the model prefer the null alignment over links between actual words. Note that the variational approach worsens this problem of the NULL token as mentioned above. We discuss BPE-based alignment in Chapter 6.

- Our variants help to greatly improve alignment links for rare/unknown. This again proves the benefit of using the subword units and also of the reconstruction component in VAEs.

- The benefits of our variants for long sentences and function/content words are less clear when compared with the vanilla variational models.

- Another difficult problem with this model is to control the optimization. It is a difficult task when the objective functions combine multiple terms with varying dynamics. More work is needed there to design better optimization strategies, with a better balance between the various sub-objectives.

- The symmetrical alignment problem is still far from solved. We still need a model more symmetrical. Let's recall our parameter sharing approach where we simultaneously train the alignment models in both directions and they use the same decoder respectively for f_1^J and e_1^I. Therefore, sharing information between the two encoders would make the model even more symmetrical. One possible solution is to encode both source and target sentence by using only one encoder.

- A more complex decoder using RNNs or contextual architecture is also an area that we should explore. However, this requires a good strategy for optimization to eliminate the problem of posterior collapse.

We highlight again that using a subword tokenization algorithm namely BPE failed to create a large number of one-to-many or many-to-many links. Therefore, we will explore behaviors of BPE-based alignments in Chapter 6.

Chapter 6

Using subwords in word alignments

State of the art open vocabulary neural machine translation systems are based on subword units which help to handle unknown words or rare words. Several algorithms used to generate subword units are BPE [Sennrich et al., 2016], WordPiece [Wu et al., 2016] and Unigram Language Model [Kudo, 2018]. For the task of word alignment, this also helps to produce finer-grained alignments, i.e. alignments between morphemes or language features (Section 2.3.2). We saw a remarkable improvement for rare/unknown words using BPE-based vocabulary of size equal to 32K compared with word-based/character-based models (Section 5.4.7). Note that vocabulary size controls the trade-off between character level and word level tokenization [Burlot and Yvon, 2017]. However, the choice of vocabulary size is generally made by the following existing recipes. Huck et al. [2017] design a linguistically-informed segmentation techniques by looking at the shortcomings of BPE segmentations. Ding et al. [2019a] conduct a systematic exploration with various numbers of BPE merge operations to understand its interaction with NMT system performance. They mainly compare several NMT architectures such as shallow/deep-transformer, tiny/shallow/deep-LSTM and report BLEU scores. Bostrom and Durrett [2020] evaluate the impact of tokenization on language model pre-training. They conclude that tokenization encodes a surprising amount of inductive bias and LM tokenization produces subword units that qualitatively align with morphology much better than those produced by BPE. Therefore, they suggest that unigram LM tokenization may be the better choice than BPE tokenization for the development of pretrained models.

In this chapter, we explore how different BPE configurations affect word alignment performance and propose a recommendation for selecting proper BPE configurations for our six language pairs. Therefore, we make the following contribution:

- A systematic comparison of several BPE configurations points out their benefits and limitations for the alignment task. We not only report AER, F-score, recall and precision, but also discuss the issues of rare words, alignment types, sequence lengths and symmetrization.
- We establish a proper BPE configuration for each language pair for further studies.

We first describe our experiments in Section 6.1. Performance of different BPE configurations are displayed in Section 6.3. Rare words and alignment types are respectively discussed in Section 6.5 and Section 6.4. We explore the issue of sequence lengths in Section 6.2. Our final analysis is about symmetrization (Section 6.6). Complete results are in [Ngo Ho, 2021, Appendix E].

Contents

6.1 Experiments

We perform the alignment between subword units generated by Byte-Pair-Encoding [Sennrich et al., 2015], implemented with the SentencePiece model [Kudo and Richardson, 2018]. All parameters of this model are set to their default values. We independently segment sentences in each language with different vocabulary sizes $V \in [2K, 4K, 8K, 16K, 32K, 48K]$. For Japanese, we do not use the vocabulary size of 2K because it is smaller than the character-based vocabulary size. For English-Vietnamese, experiments for English vocabulary size of 48K and Vietnamese vocabulary size of 32K and 48K were not performed. This is because they are larger than their word-based correspondences (Section 3.1.1).

Subword-level alignments are converted into word-level alignments as follows: a link between a source and a target word exists if there is at least one link alignment between their subwords (Section 2.3.2). An example of a BPE-based sentence for different vocabulary sizes is displayed in Figure 6.1[1].

We distinguish between two conditions:

- Small vocabulary size (i.e., 2K, 4K and 8K): In these cases, there are more short tokens, which lengthens sequences. We expect that this would help to generate more links after using the recombination algorithm.

- Large vocabulary size (i.e., 32K and 48K): Larger vocabulary size makes a sequence of subwords more similar to a sequence of words.

We use `Fastalign` and `Eflomal` for this alignment task with a large number of jobs (i.e., about 36 jobs for each language pair) since it is a simple and computationally efficient tool. Note that we concatenate training and test data. Complete results are shown in [Ngo Ho, 2021, Appendix E].

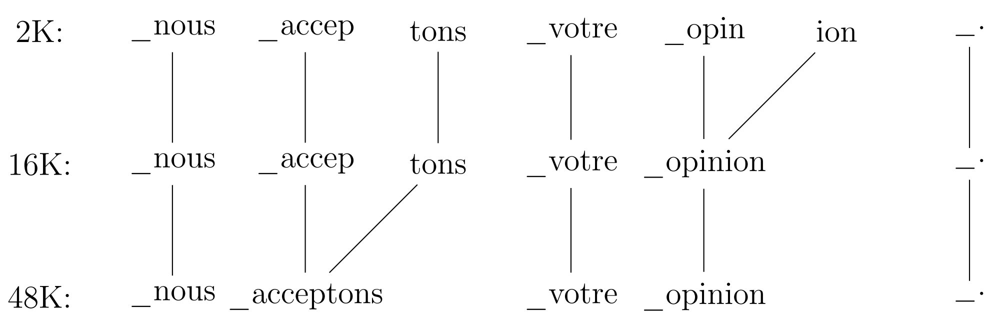


Figure 6.1: Example of a BPE-based sentence for different vocabulary sizes of 2K, 16K and 48K

[1]For the SentencePiece, the subword beginning a word starts with an underscore e.g., "_nous _accep tons".

6.2 Sequence lengths for BPE level and word level

The use of subwords often lengthens input sequences, which can be harmful to model performance. In order to check if this is an issue, we plot the alignment scores as a function of length difference between word-based sequence and BPE-based sequence [Ngo Ho, 2021, Appendix E.10]. Note that we take the mean value for each length difference. As can be seen in Figure 6.2, shortening tokens (e.g., using a vocabulary size of 2K-4K) can lead to the length difference of nearly 30 tokens (in English) and 50 tokens (in German). We also observe that that larger length differences (e.g., 2K) clearly makes the alignment task more difficult. The worse AER (about 60%) is observed in the case of 2K-2K.

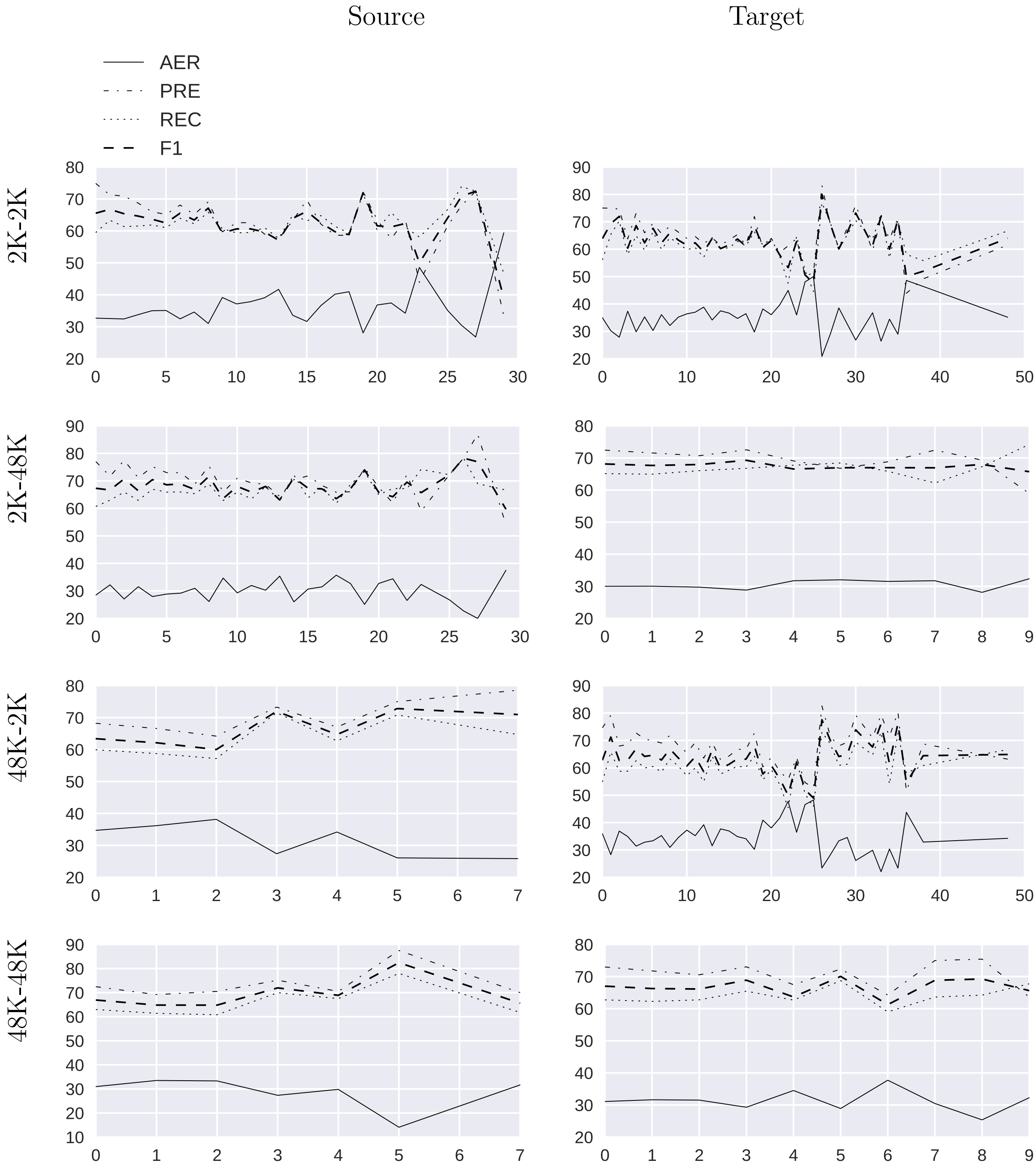


Figure 6.2: BPE-based `Fastalign` for English-German: Alignment error rate (AER), F-score (F1), precision (PRE) and recall (REC) as a function of the length difference. To compute the length difference, we subtract a word-based sentence length from a BPE-based sentence length.

We also observe AER scores as a function of sentence length difference (i.e., subtracting the length of the target sentence from the length of the source sentence), shown in [Ngo Ho, 2021, Appendix E.11]. We can see similar trends between word-based (Section 3.10) and BPE-based alignment: smaller length differences often obtain better AER scores as can be seen in Figure 6.3.

Figure 6.3: The direction English-Japanese: AER score as a function of sentence length difference. The x-axis shows the sentence length difference. The y-axis represents the AER. The difference is computed by subtracting the length of the target sentence from the length of the source sentence.

6.3 Do different BPE-based vocabulary sizes make different alignment patterns?

In order to observe how the alignment accuracy varies with the size of the BPE vocabulary, we plot AER, precision, recall and F-score as a function of the target vocabulary size for each source vocabulary size in [Ngo Ho, 2021, Appendix E.1.1]. Moreover, we also show a comparison between BPE-based and word-based scores in [Ngo Ho, 2021, Appendix E.1.2].

The first observation is that short units in both sides always yield a better recall. For example in Figure 6.4, the top-left zone of recall contains the best scores. In fact, shorter BPE units on the source side help to generate more one-to-many/many-to-many links. This can be seen in Figure 6.15, the higher numbers of correct one-to-many/many-to-many links are found for the smaller source vocabulary sizes. We can also observe this trend for unaligned words in [Ngo Ho, 2021, Appendix E.3].

In addition, we observe that AER and precision often share similar patterns (except for English-Vietnamese). An explanation is proposed in Section 3.2.

Several additional observations can be made:

- In Figure 6.4, the best AER scores and precision are in the zone (bottom-right) of large vocabulary sizes on both sides. However, this zone has the worse recall. This means that short BPE units improve recall but hurt precision. Remind that we see the same problem for word-based alignment where models favor precision over recall (Section 3.2). In Figure 6.5, we can see that the precision scores of the largest vocabulary size (i.e., 48K) unsurprisingly are similar to the word-based alignment. Similar trends are found in the other corpora, namely English-German and English-Czech.

- We notice the case of English-Romanian where short units in the target Romanian side (the bottom-left zone) yield an improved AER and precision (see Figure 6.6). We observe alignment types generated in this zone (large source vocabulary sizes i.e., 48K and small target vocabulary sizes) in Figure 6.7. For the direction English-Romanian, the number of alignment links decreases because of the large source vocabulary size. However, the model still keeps a large number of many-to-one links since short units in the target Romanian side tend to generate more links belonging to this alignment type. In the opposite direction where English is on the target side, the effect of short units is less clear because most links in the bottom-left zone are one-to-one links.

- In the direction Japanese-English (Figure 6.8), alignment between short Japanese units yields the better AER, precision, recall and also F-score. An explanation for this agreement between these measures is that there are not "possible" links, which means that favoring precision over recall does not help to get a better AER. In the direction English-Vietnamese (Figure 6.9), there is a mismatch between precision and the other scores (AER, recall, F-score). This is because the gain in recall is larger than the corresponding loss in precision.

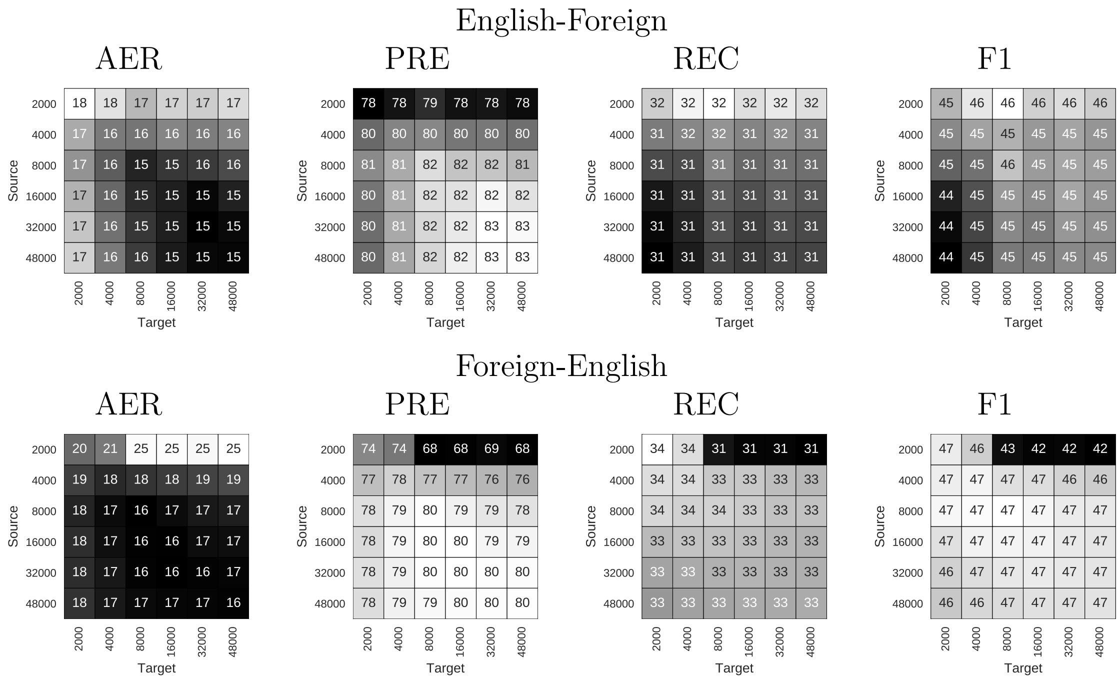


Figure 6.4: BPE-based `Fastalign` for English-French: For each pair (vocabulary size of source and target), we display Alignment error rate (AER), F-score (F1), precision (PRE) and recall (REC).

We recheck these findings with `Eflomal` [Östling and Tiedemann, 2016]. In general, `Eflomal` obtains a better performance than `Fastalign`. There are several small disagreements about the best target vocabulary size, for the language pairs English-Czech and English-Romanian (in both directions) and for the direction French-English. We observe the performance of the two models for the direction English-Romanian in Figure 6.10. In the recall matrix, better performance is found in the top zone (small vocabulary sizes) for both models. The best score is achieved by the vocabulary size pair 2K-4K for `Fastalign` and by 4K-48K for `Eflomal`. An explanation is that using BPE in the source side affects the number of links more importantly than in the target side. In fact, we obtain these symmetrical results from asymmetrical alignments. We notice another disagreement in precision for the direction English-Vietnamese (Figure 6.11) which requires further studies. The best pair for `Fastalign` is 32K-16K whereas the pair 2K-16K gets the best performance for `Eflomal`. This small difference does not create any change for the F-score.

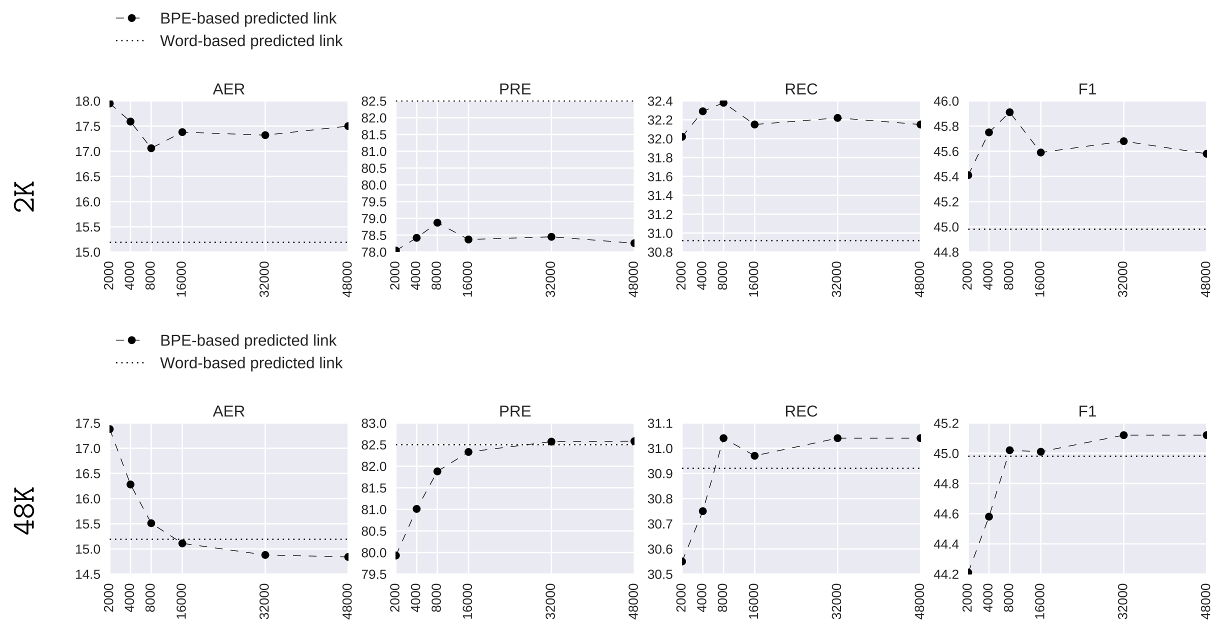


Figure 6.5: BPE-based `Fastalign` for the direction English-French: For each source vocabulary size, we display Alignment error rate (AER), F-score (F1), precision (PRE) and recall (REC) as a function of the target vocabulary size.

Best vocabulary size pairs Table 6.2 reports the pair of source and target vocabulary size that obtain the best score (in parentheses) for each performance measure. More details about these scores are in [Ngo Ho, 2021, Appendix E.1.2]. We see that the benefit of using BPE is less clear when English is in the target side e.g., French-English, Romanian-English and Czech-English. Moreover, for French, German, Czech and English, the gains are maximal when we use large vocabulary sizes (e.g., 32K). An explanation is that too short BPE units can cause the loss of important information regarding words. For Romanian, Japanese and Vietnamese, a small vocabulary size (e.g., 4K and 8K) is an appropriate choice. This is because generating more links is very helpful to increase the performance. For each language pair, we suggest the best BPE configurations found in our experiments:

- English-French: We see that 32K word vocabulary for French and 16K word vocabulary for English obtain the best AER and precision. The best F-score and recall are made by small vocabulary sizes e.g., 8K-8K/16K. Recall that this language pair has a large number of fuzzy links, using 32K can reduce number of links. However, the vocabulary size of XX-32K still helps to get a better F-score and recall than the word-based model. Therefore, the use of 32K word vocabulary for French is not a bad choice. We prefer 8K-16K because of the balance between precision and recall.

- English-German: In the direction English-German, the source and target vocabulary sizes for English should be respectively higher than 4K and 16K to gain better scores than the word-based model. The best English and German vocabulary size is respectively 4K and 32K. In the opposite direction, the best English vocabulary size that helps to outperform word-based models is 16K.

- English-Romanian: We again see the benefit of using 16K for English vocabulary size on source side. In the opposite direction, BPE for `Fastalign` fails to generate better performance than the word-based model. However, using heuristic symmetrization (i.e., GDF) helps to gain some more points, and to outperform this word-based model (Section 6.6).

- English-Czech: the 16K English word vocabulary still helps to achieve the best performance. We also see in the case of `Fastalign` that AER, recall and F-score can be easily improved except for precision. The best pair is 48K-48K in precision which only gives a

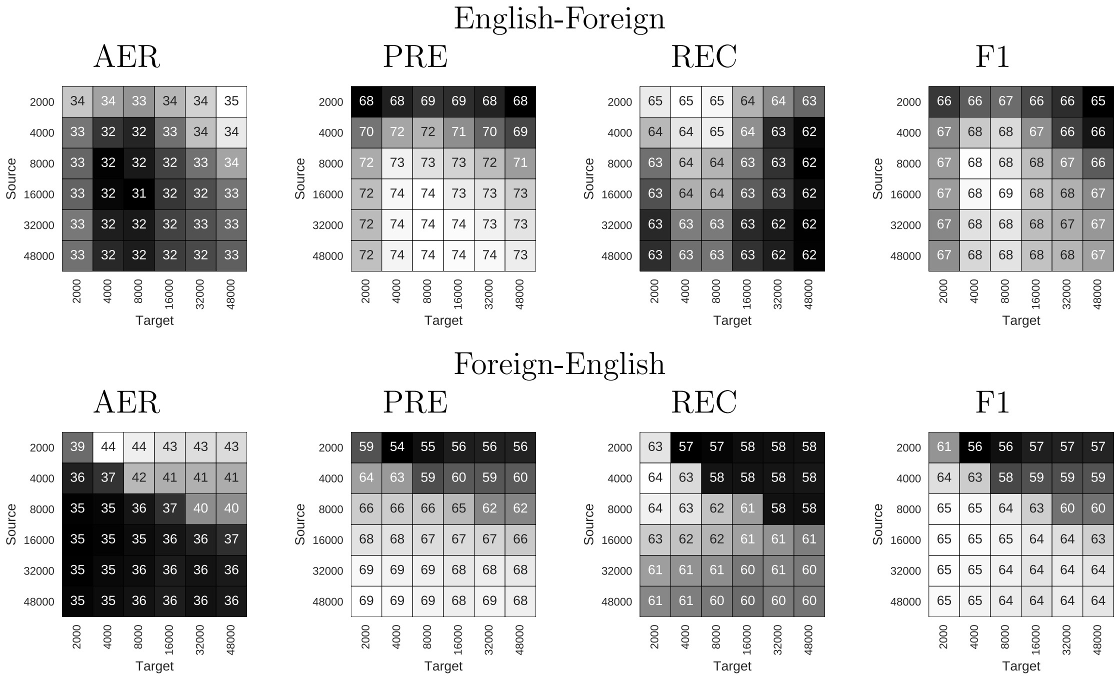


Figure 6.6: BPE-based `Fastalign` for English-Romanian: For each pair (vocabulary size of source and target), we display Alignment error rate (AER), F-score (F1), precision (PRE) and recall (REC).

slight gain compared with the word-based model in the direction English-Czech. Note that using GDF does not help to outperform the word-based model in precision (Section 6.6). In general, the Czech vocabulary size of 32K is an appropriate configuration.

- English-Japanese: Best scores are achieved by small vocabulary sizes (e.g., 4K). The best vocabulary size for Japanese is 8K while for English it can be larger e.g., 16K or 32K

- English-Vietnamese: We see similar behaviors as mentioned in English-Japanese: Small vocabulary sizes still work well for the English-Vietnamese language pair. The best parameter configuration is 2K-8K in the direction English-Vietnamese and 2K-32K in the opposite direction. Keeping short Vietnamese units can yields more links, which helps to cover a large number of many-to-one links (Table 3.8) where several Vietnamese words align with one English word.

AER and large vocabulary sizes We observe in detail how large vocabulary improve precision and AER score. We collect correct and incorrect alignment links (Section 3.3). Figure 6.12 displays the alignment links for English-Japanese.

- Compared with the word-based model, most BPE vocabulary size pairs increase correct alignment links, except for 48K-4K, 32K-4K and 16K-4K. These exceptions are in the case where longer BPE units align with short BPE units. We also notice that these pairs fail to reduce incorrect non-alignment links. An explanation is that short units in Japanese side suffer a loss of information regarding words. English BPE units (being close to word-level units because of their large vocabulary sizes) align with incorrect Japanese units. These incorrect units can be the most frequent words.

- Large target vocabulary sizes (e.g. *-48K) favor NULL links with a large number of TN. This means that long BPE units helps models to distinguish between non-aligned token and other words.

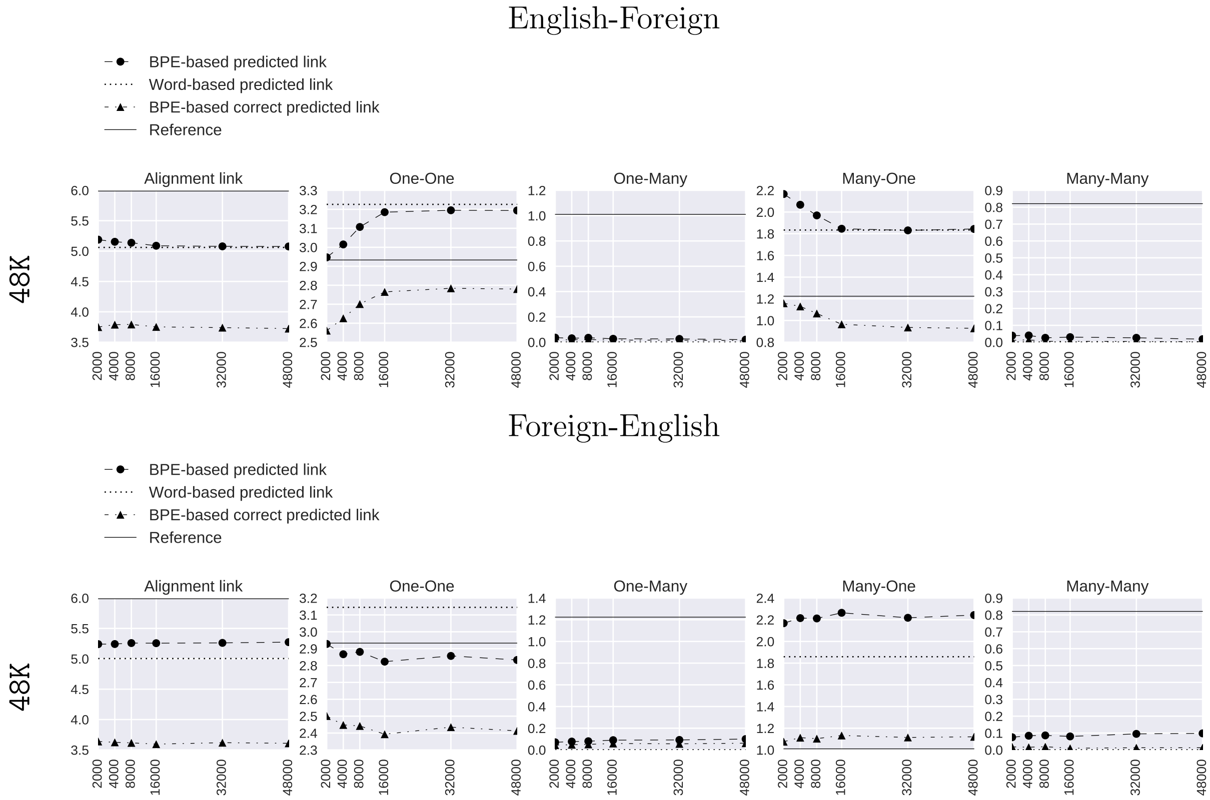


Figure 6.7: BPE-based `Fastalign` for English-Romanian: We observe the alignment types. For each source vocabulary size, we show number of links as a function of the target vocabulary size. The y axis corresponds to the number of links ($\times 1000$).

Model	Test corpus	AER	PRE	REC	F1
`Fastalign`	English-French	32K-**32K** (14.77)	32K-**32K** (82.63)	**2K**-8K (32.38)	2K-8K (45.91)
	French-English	8K.8K (16.35) *	32K-**16K** (79.68) *	**2K**-2K (34.20)	**8K**-8K (47.22)
`Eflomal`	English-French	16K-**32K** (6.16)	16K-**32K** (92.56)	**2K**-32K (34.59)	4K-16K (49.62)
	French-English	32K.16K (7.75)	48K-**16K** (90.06)	**2K**-32K (37.57)	**8K**-16K (52.09)
`Fastalign`	English-German	**4K**-**32K** (26.71)	4K-32K (72.37)	**2K**-16K (69.84)	**4K**-**32K** (70.94)
	German-English	16K-**16K** (29.32)	**48K**-48K (71.25)	**2K**-4K (67.20)	16K-**16K** (68.38)
`Eflomal`	English-German	**4K**-**32K** (20.7)	48K-48K (83.08)	**2K**-32K (74.09)	**4K**-**32K** (76.82)
	German-English	32K-**16K** (21.79)	**48K**-32K (82.71)	**2K**-8K (72.54)	32K-**16K** (75.58)
`Fastalign`	English-Romanian	**16K**-8K (31.49)	32K-**8K** (73.82)	2K-4K (64.71)	**16K**-8K (68.53)
	Romanian-English	16K.2K (35.02) *	48K.2K (69.43) *	4K-2K (63.89)	16K.2K (65.0) *
`Eflomal`	English-Romanian	**16K**-48K (24.47)	48K-**8K** (89.09)	4K-48K (66.12)	**16K**-48K (75.55)
	Romanian-English	8K-48K (24.53)	32K-48K (84.72)*	8K-48K (68.45)	8K-48K (75.49)
`Fastalign`	English-Czech	16K-**32K** (24.60)	48K-**48K** (71.01)	**2K**-4K (62.96)	8K-16K (65.72)
	Czech-English	32K-**16K** (24.33)	**48K**-16K (72.34) *	2K-4K (61.62)	16K-**16K** (64.54)
`Eflomal`	English-Czech	8K-**32K** (12.56)	32K-**48K** (87.1)	**2K**-32K (64.55)	4K-32K (73.25)
	Czech-English	48K-**16K** (11.91)	**48K**-4K (89.14)	4K-48K (64.43)	8K-**16K** (73.61)
`Fastalign`	English-Japanese	**8K**-8K (47.51)	8K-**16K** (57.31)	**8K**-8K (48.78)	**8K**-8K (52.49)
	Japanese-English	**8K**-16K (46.91)	**8K**-48K (56.62)	**4K**-16K (51.34)	**8K**-16K (53.09)
`Eflomal`	English-Japanese	**8K**-32K (42.5)	32K-**16K** (65.63)*	**8K**-32K (51.5)	**8K**-32K (57.5)
	Japanese-English	**8K**-32K (41.75)	**8K**-32K (64.14)	**4K**-8K (54.43)	**8K**-32K (58.25)
`Fastalign`	English-Vietnamese	4K-4K (45.74)	32K-16K (57.43)	**2K**-2K (54.50)	4K-4K (54.27)
	Vietnamese-English	4K-8K (29.52)	8K-16K (67.20)	**2K**-8K (74.48)	4K-8K (70.48)
`Eflomal`	English-Vietnamese	2K-8K (36.19)	2K-8K (66.84)	**2K**-4K (61.08)	2K-8K (63.82)
	Vietnamese-English	2K-32K (24.96)	4K-32K (75.53)	**2K**-32K (74.73)	2K-32K (75.05)

Table 6.2: `Fastalign` and `Eflomal`: The best pair of source and target vocabulary sizes for each performance measure i.e., Alignment error rate (AER), F-score (F1), precision (PRE) and recall (REC). Note that * means the word-based model gets the best score.

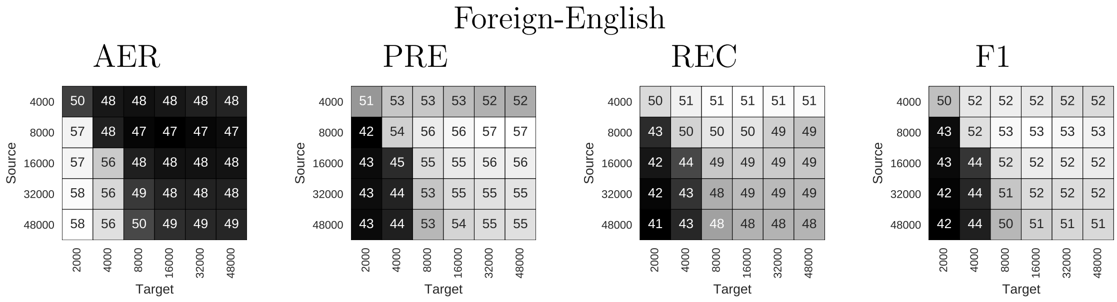


Figure 6.8: BPE-based `Fastalign` for the direction Japanese-English: For each pair (vocabulary size of source and target), we display Alignment error rate (AER), F-score (F1), precision (PRE) and recall (REC).

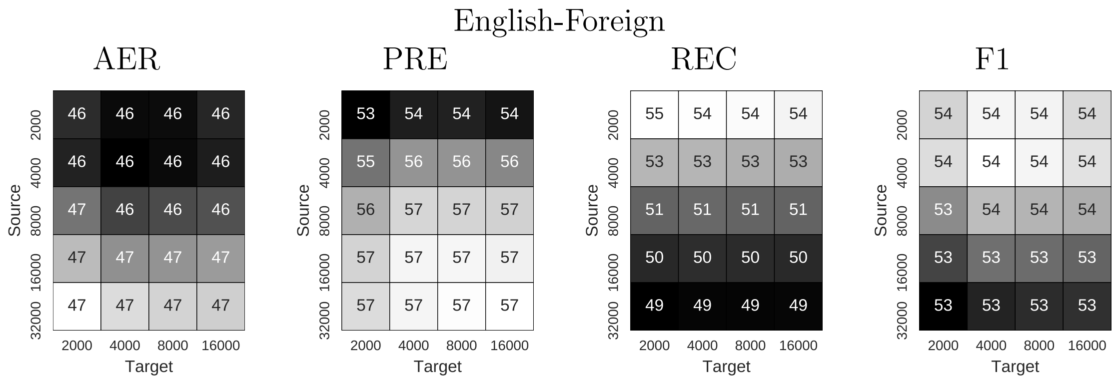


Figure 6.9: BPE-based `Fastalign` for the direction English-Vietnamese: For each pair (vocabulary size of source and target), we display Alignment error rate (AER), F-score (F1), precision (PRE) and recall (REC).

- The pair 8K-8K obtains the best AER score. It helps to predict more correct alignment links (better than the pair *-48K) and also more correct NULL links (better than than the pair *-4K).

Similar trends are also observed in other language pairs and in both directions. Complete results are in [Ngo Ho, 2021, Appendix E.2].

Unaligned words and recall We observe unaligned words to understand more precisely the effect of an improved recall for shorter units and a rise of NULL links for longer units. Complete results are in [Ngo Ho, 2021, Appendix E.3]. As can be seen in Figure 6.13 (English-Japanese), we see that large vocabulary sizes tend to generate more non-alignment links. The gain for correct unaligned words is much larger than the corresponding loss for incorrect unaligned words. Take a closer look at the cases 8K-(8K, 16K, 48K), larger target vocabulary size (8K-16K and 8K-48K) greatly increase the number of incorrectly unaligned words while still keep similar numbers of correct unaligned words. In addition, the number of correct unaligned words for the case 8K-8K is larger than in the case 8K-4K. This helps the pair 8K-8K to get the best performance.

Using the regularization method BPE-dropout We explore the benefit of using BPE-dropout in [Ngo Ho, 2021, Appendix E.1.3]. For each sentence, we generate five different BPE-based sentences, which leads to a larger training corpus. Figure 6.14 displays the performance of `Fastalign` with/without BPE-dropout. We can observe a small gain (+1) in recall but a larger loss in precision and AER. In general, we do not see a clear improvement of BPE-dropout for the word alignment task. Similar trends are found for other language pairs and for other directions.

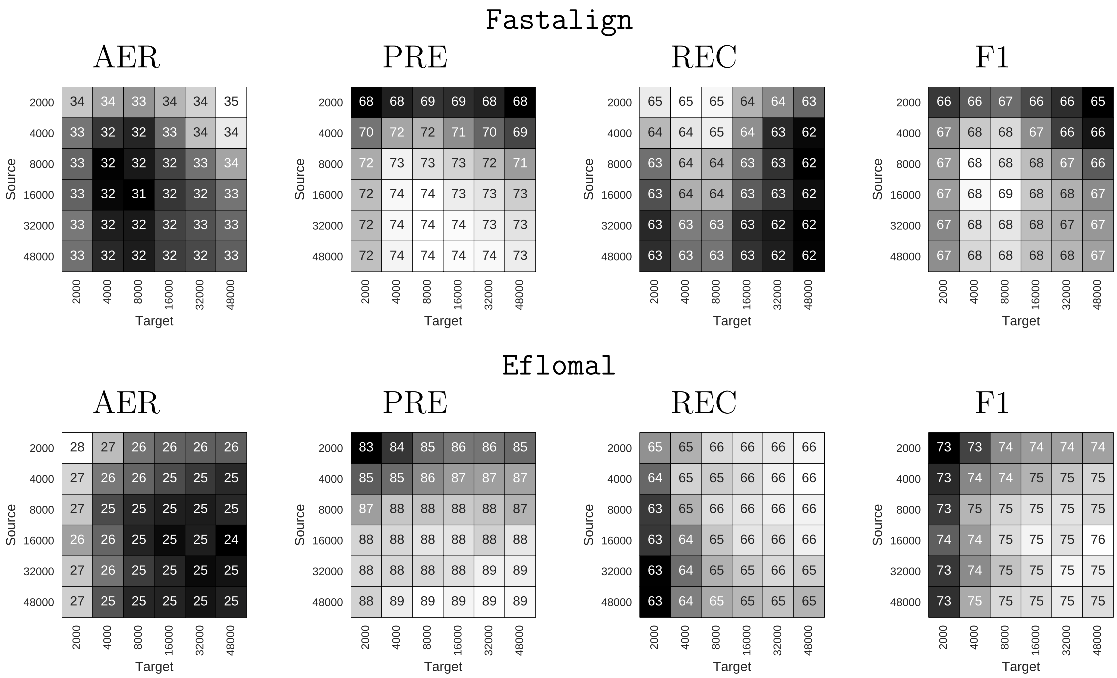


Figure 6.10: The direction English-Romanian: For each pair (vocabulary size of source and target), we display Alignment error rate (AER), F-score (F1), precision (PRE) and recall (REC) for `Fastalign` and `Eflomal`.

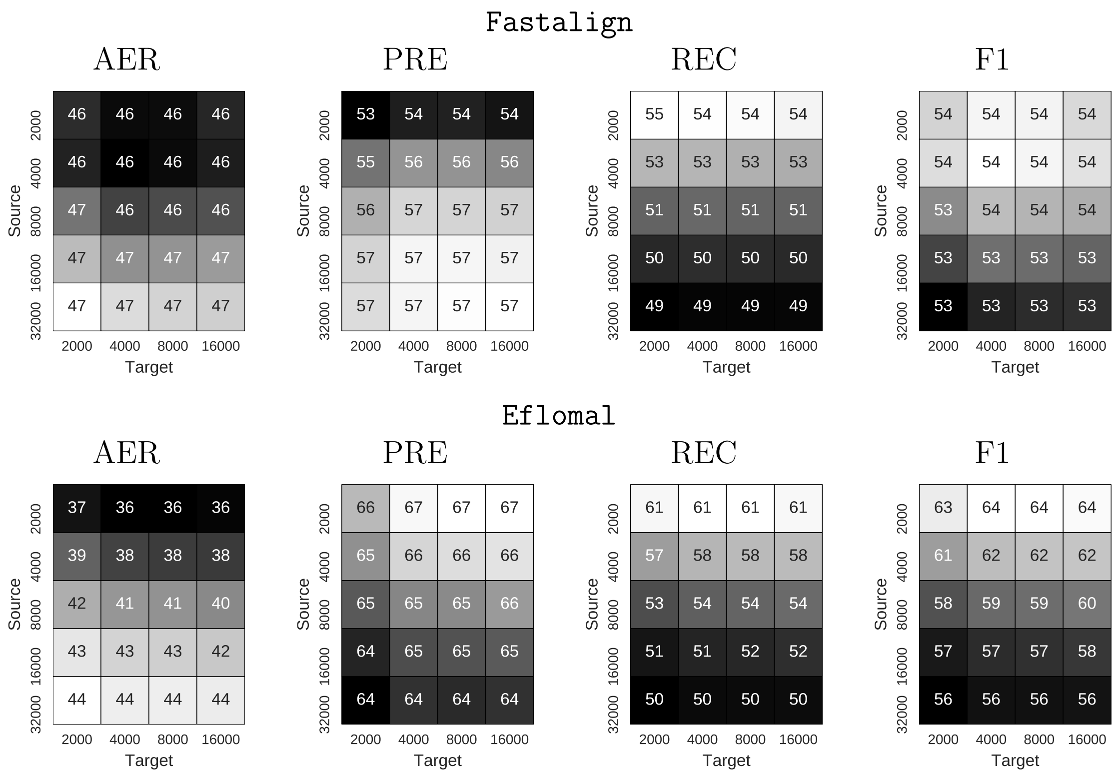


Figure 6.11: The direction English-Vietnamese: For each pair (vocabulary size of source and target), we display Alignment error rate (AER), F-score (F1), precision (PRE) and recall (REC) for `Fastalign` and `Eflomal`.

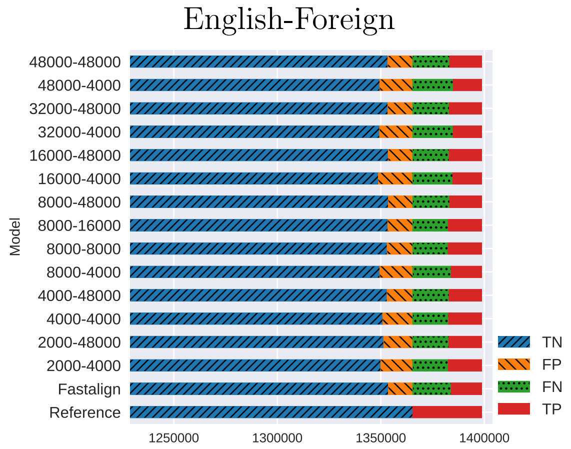


Figure 6.12: BPE-based `Fastalign` for English-Japanese: We observe correct and incorrect alignment links.

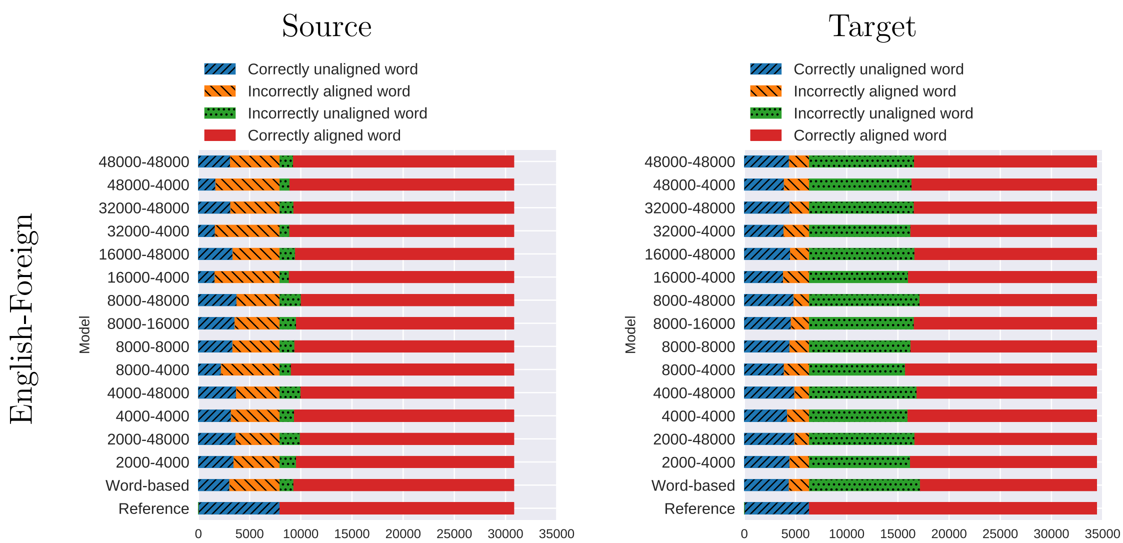


Figure 6.13: BPE-based `Fastalign`: Unaligned words for the direction English-Japanese

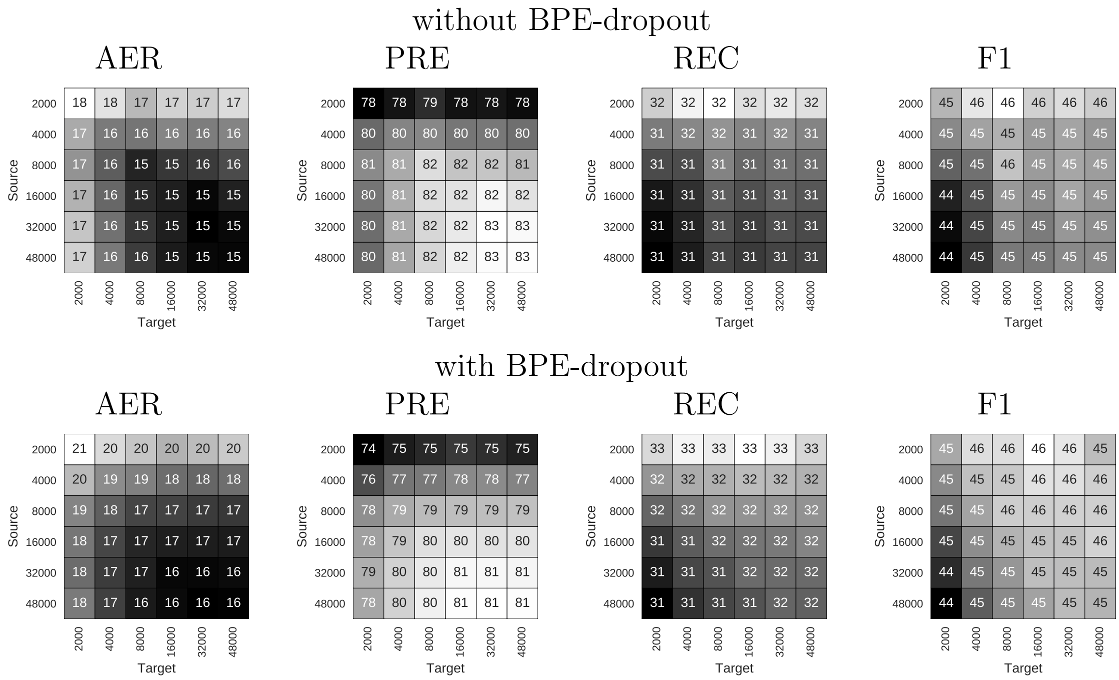


Figure 6.14: BPE-based `Fastalign` with/without BPE-dropout for the direction English-French: For each pair (vocabulary size of source and target), we show Alignment error rate (AER), F-score (F1), precision (PRE) and recall (REC).

6.4 One-to-one and many-to-many links

We discuss in detail how the number of links for each alignment type changes according to the vocabulary size. Complete results are in [Ngo Ho, 2021, Appendix E.4]. We observe the two extreme values of source vocabulary size i.e., 2K and 48K for English-German (see Figure 6.15). The most noticeable observation is that shorter BPE units eventually generate fewer one-to-one links and more links for the other alignment types, especially one-to-many and many-to-many. In other words, a token that decomposes into a sequence of shorter tokens in the source side has more chance to align with several target tokens. However, we do not see a large number of correct one-to-many/many-to-many links.

- The top graphs (the smallest source vocabulary size i.e., 2K): the number of one-to-one links gradually increases for the small target vocabulary sizes (from 2K to 8K). The opposite trend is found in other alignment types. Note that the differences between the large target vocabulary sizes (e.g, 16K, 32K and 48K) are significant for only many-to-many links, for which we see a clear decrease when the number of BPE units increases. However, the number of correct one-to-many links remains unchanged from 2K to 48K. In addition, only a small number of many-to-many links is correct e.g., about 300 correct links vs 1200 links in the case of 2K word target vocabulary.

- Varying the source vocabulary from 2K to 48K, we see that the main changes mostly affect one-to-many and many-to-many links.

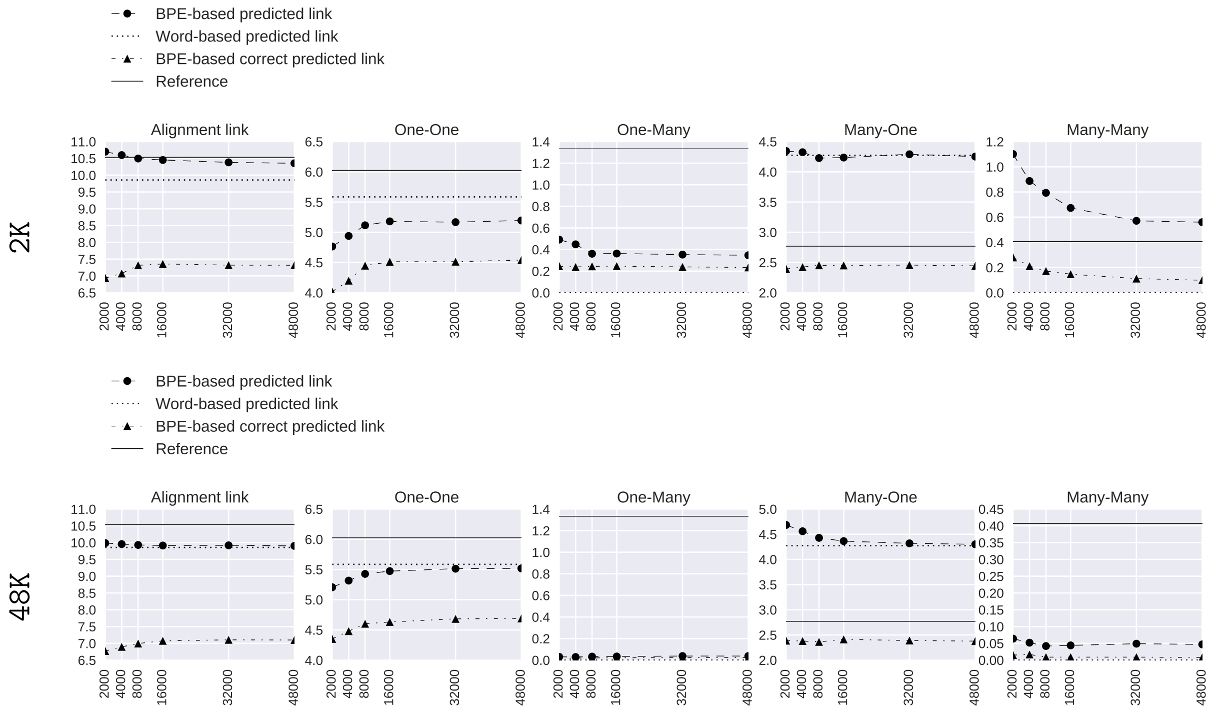


Figure 6.15: BPE-based `Fastalign` for the direction English-German: We observe the alignment types. For each source vocabulary size, we show the number of links as a function of the target vocabulary size. The y axis corresponds to the number of links ($\times$1000).

6.5 Rare words in BPE-based alignments

Using subwords eliminates unknown words and reduces the problem of rare words. To observe gains of using subwords for the word alignment task, we plot AER, precision, recall and F-score of rare source/target words as a function of the target vocabulary size for each source vocabulary size. Complete results are in [Ngo Ho, 2021, Appendix E.5].

We observe that all language pairs benefit from subword alignments with the best F-scores. However, for the direction Czech-English, Romanian-English and English-Japanese (both directions), all BPE-based models still lag behind the word-based model in precision. This is simply because word-based models generate fewer alignment links than BPE-based models. In Figure 6.16, we show the result for the vocabulary pair 16K-16K for the language pair Czech-English. Despite the higher precision for word-based alignment, the number of correct links for BPE-based alignment is still larger.

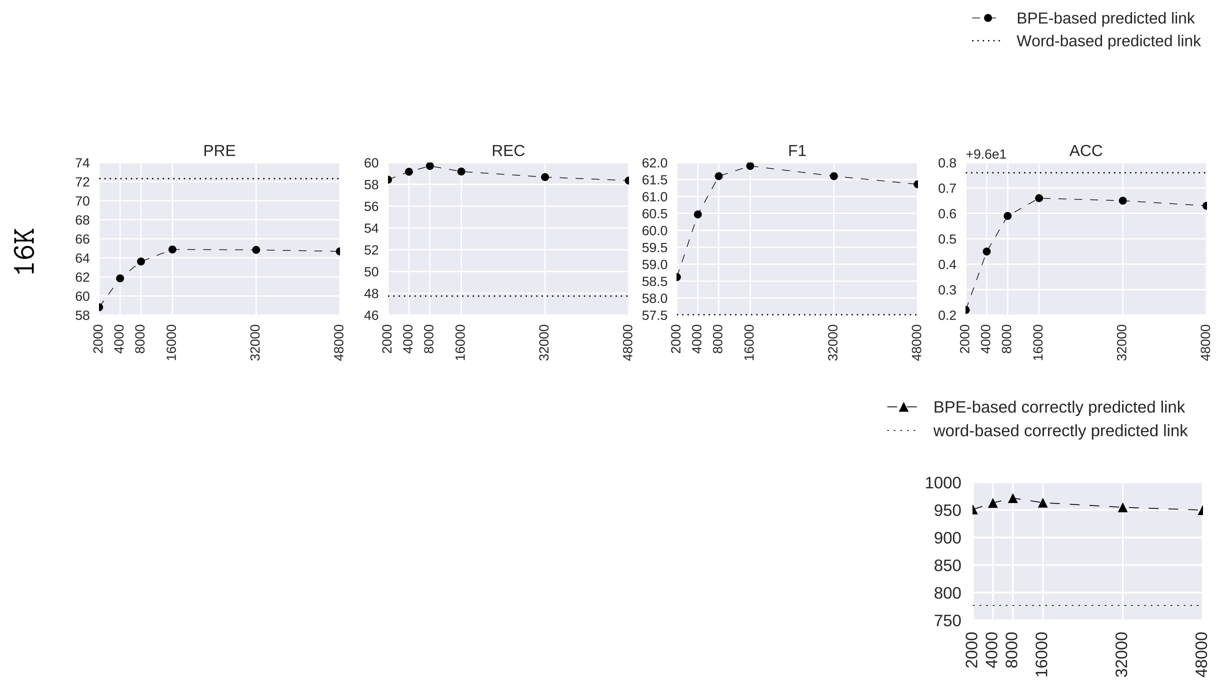


Figure 6.16: BPE-based `Fastalign` for the direction Czech-English: In the four top graphs, we observe the scores for rare source words. For each source vocabulary size, we report the accuracy (ACC), F-score (F1), precision (PRE) and recall (REC) as a function of the target vocabulary size. The bottom graph shows the number of correct links for rare source words.

We observe the scores for rare German words in Figure 6.17. Recall that German has a very large word-based vocabulary size (Table 3.2). In the direction English-German, we can see a large gain (+3 scores) for F-score when using the German vocabulary size of 32K/48K. For the opposite direction, using only 16K BPE-units vocabulary for German can reach about 71 F-score, better than 56.44 F-score obtained by using ∼300K word vocabulary.

Moreover, we try to explain why the recall for rare words using BPE tokenization are larger than using word-based models. Figure 6.18 displays the average number of BPE-based fragments as a function of word occurrence in two cases: 2K and 48K vocabulary size[2]. Recall that less frequent words often have a greater length (Section 3.7). Therefore, for the 2K word vocabulary, we can see that these words often decompose into more fragments, leading to a larger number of links and to a higher recall for rare words as mentioned above. This observation is less clear for 48K. In Figure 6.19, we observe the number of one-to-many/many-to-one as a function of word occurrence in two cases: 2K and 48K vocabulary size[3]. Less frequent words often generate more one-to-many/many-to-one links. In the case 2K-2K, we can see a clear

[2]Complete results are in [Ngo Ho, 2021, Appendix E.6]. Note that figures only display word occurrence smaller than 1000.

[3]Complete results are in [Ngo Ho, 2021, Appendix E.7]. Note that figures only display word occurrence smaller than 100.

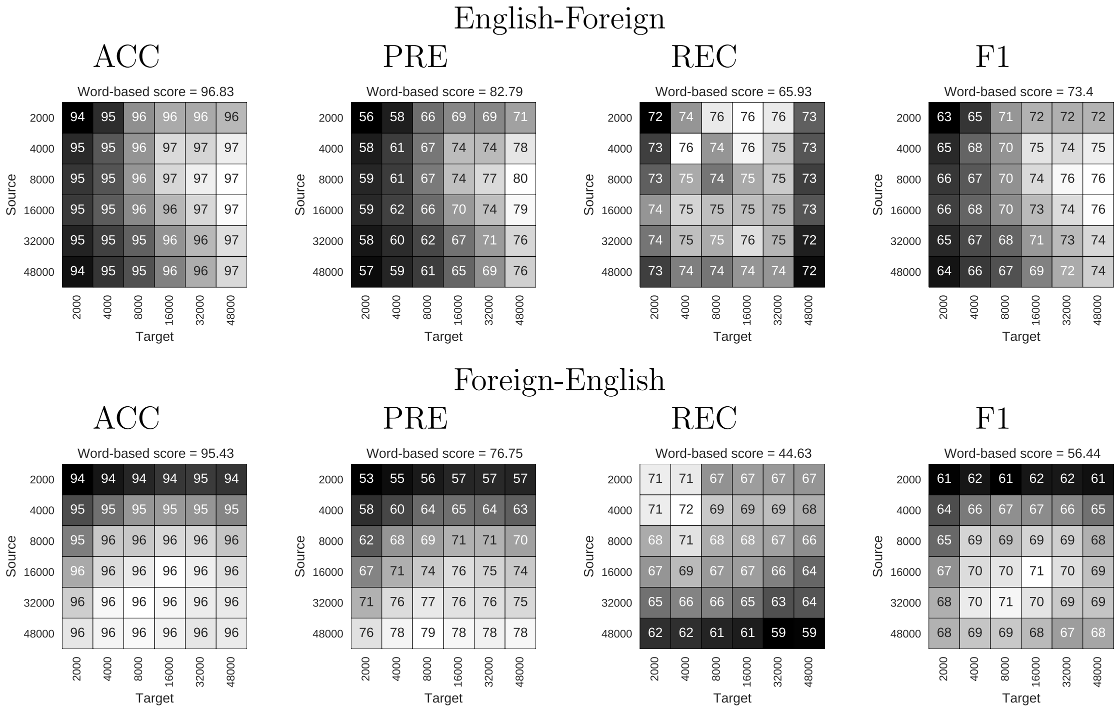


Figure 6.17: BPE-based `Fastalign`: We observe the scores for rare German words in both directions English-German and German-English. For each source vocabulary size, we show the accuracy (ACC), F-score (F1), precision (PRE) and recall (REC) as a function of target vocabulary size.

difference where the number of predicted one-to-many/many-to-many links is often larger than the number of corresponding reference links.

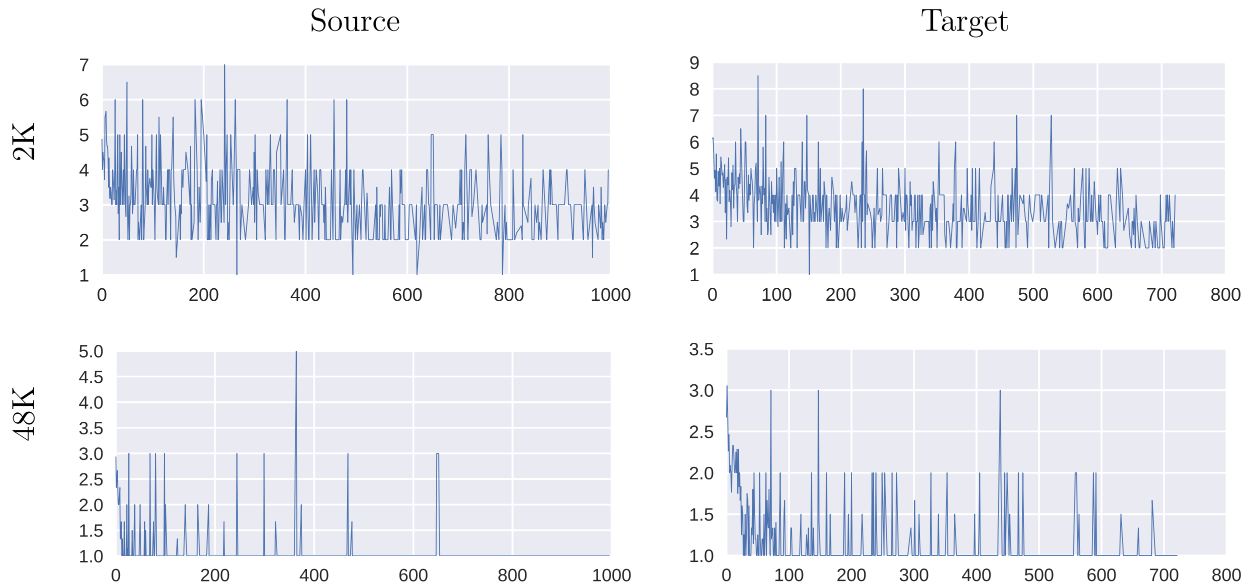


Figure 6.18: The direction English-German: Average number of BPE-based fragments as a function of word occurrence.

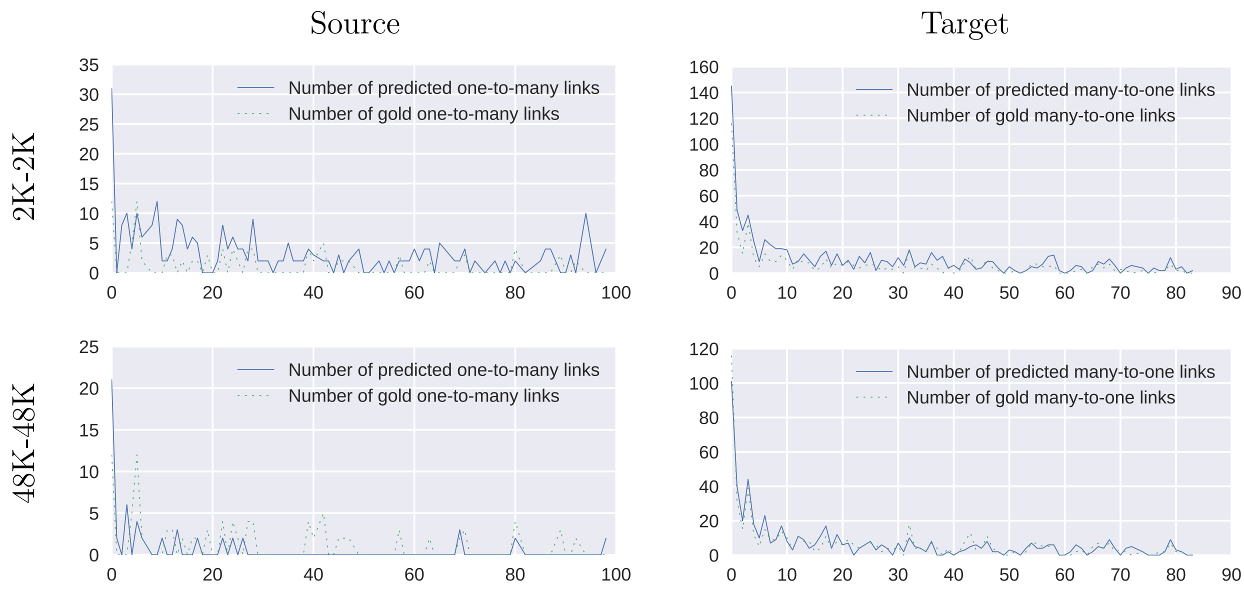


Figure 6.19: The direction English-German: Number of one-to-many (left graphs) and many-to-one (right graphs) links as a function of word occurrence.

6.6 Symmetrizing subword based alignments

Heuristic symmetrization e.g., grow-diag-final is an important post-process to obtain better alignments. When using BPE units, we consider two options:

- GDF-before: We first apply GDF to asymmetrical BPE-based alignments in both directions to compute a symmetrical alignment, and then transform it into the word-based alignment. Note that we only symmetrize the two asymmetrical alignments of the same BPE-based test corpus.

- GDF-after: We first transform asymmetrical BPE-based alignments into word-based alignments, and then apply GDF to the word-based alignments for obtaining the symmetrical alignments. In this methode, we can symmetrize different alignment sets with different vocabulary sizes.

Complete results are in [Ngo Ho, 2021, Appendix E.9]. We collect the best scores from both methods and the word-based model, shown in Table 6.4. As can be seen in this table, GDF-after always yields better recall, leading to better F-scores (except for English-German) than GDF-before. For the AER, Romanian, Japanese and Vietnamese benefit from this method.

Compared with the word-based models, we only see an improved AER for English-German, an better F-score for English-German, English-French and English-Czech. For other language pairs, symmetricized BPE-based alignments still lag a few points behind the word-based alignments. However, it should be noted that using BPE significantly reduces the complexity of the softmax computation that remains a problem for word-based models.

Test corpus	GDF	AER	PRE	REC	F1
English -French	Word-based	**14.25**	81.84	35.02	49.05
	Before	*32K-32K* (15.0)	**32K-32K** (83.0)	2K-8K (32.0)	2K-8K (46.0)
	After	16K-32K vs 8K-8K (17.0)	16K-32K vs 8K-8K (77.0)	**2K-8K vs 32K.48K** (38.0)	**16K-32K vs 8K-2K** (50.0)
English German	Word-based	28.21	70.12	69.57	69.84
	Before	**4K-32K** (27.0)	**4K-32K** (72.0)	2K-16K (70.0)	**4K-32K** (71.0)
	After	4K-32K vs 4K-16K (28.0)	4K-48K vs 16K-16K (66.0)	**4K-32K vs 4K-16K** (75.0)	4K-32K vs 4K-16K (70.0)
English -Romanian	Word-based	**30.42**	72.65	66.8	**69.6**
	Before	16K-8K (31.4)	**32K-8K** (74.0)	2K-4K (65.0)	16K-8K (68.52)
	After	*4K-4K vs 32K-32K* (31.0)	4K-4K vs 32K-32K (67.0)	**16K-4K vs 48K-32K** (71.0)	*4K-4K vs 32K-32K* (68.56)
English -Czech	Word-based	**23.3**	**72.95**	61.82	66.93
	Before	*16K-32K* (24.6)	*48K-48K* (71.0)	2K-4K (63.0)	8K-16K (66.0)
	After	4K-16K vs 4K-8K (25.0)	4K-16K vs 4K-8K (67.0)	**16K-32K vs 4K-4K** (68.0)	**4K-16K vs 4K-8K** (67.0)
English -Japanese	Word-based	**44.79**	**64.99**	47.99	**55.21**
	Before	8K-8K (48.0)	*8K-16K* (57.0)	8K-8K (49.0)	8K-8K (52.0)
	After	*8K-8K vs 8K-48K* (46.0)	4K-8K vs 8K-48K (52.0)	**8K-8K vs 4K-48K** (58.0)	*8K-8K vs 8K-48K* (54.0)
English -Vietnamese	Word-based	**32.9**	**64.41**	70.05	**67.11**
	Before	4K-4K (46.0)	32K-16K (57.0)	2K-2K (55.0)	4K-4K (54.0)
	After	*2K-8K vs 2K-8K* (33.0)	*32K-16K vs 4K-8K* (60.0)	**2K-8K vs 2K-8K** (76.0)	*2K-8K vs 2K-8K*(67.0)

Table 6.4: Alignment error rate (AER), F-score (F1), precision (PRE) and recall (REC) of two symmetrization methods: GDF-before and GDF-after.

6.7 Word-based, BPE-based and character-based model performance

We use the several recommended configuration of BPE-based vocabulary size (Table 6.5) for our neural models `+BPE+B` (Section 5.3), yielding the models `+BPE+B+C`. Complete results are in [Ngo Ho, 2021, Appendix E.1.5]. The first observation is that for `IBM-1`, using BPE outperforms character-based and word-based models in all language pairs. In Table 6.6, we can see that these configurations (`+BPE+B+C`) for English-German help to gain some more points of AER/F-score compared with the vocabulary size pair 32K-32K. They also outperform character-based models and word-based models. Similar trends of `IBM-1+BPE+B` and `IBM-1+BPE+B+C` are found for other language pairs/both directions.

For `HMM`, BPE-based models still lag a few points behind character-based for the language pairs English-French and English-Romanian (both directions), for the directions Japanese-English and English-Vietnamese. We observe `HMM` variants for the language pair English-Vietnamese in Table 6.7. BPE-based models obtain a better recall but a worse precision than character-based models. This loss in precision obstructs the BPE-based model performance. Recall that our neural models has a problem of over-generating null links. Using character-based models seems to be a better approach of reducing null links than using BPE-based models, especially for the language pairs English-French and English-Romanian (both directions).

Language pair	En-XX	XX-En
English-French	16K-32K	32K-16K
English-German	4K-32K	32K-16K
English-Romanian	16K-8K	8K-48K
English-Czech	16K-32K	32K-16K
English-Japanese	16K-8K	8K-16K
English-Vietnamese	2K-8K	2K-32K

Table 6.5: Several recommended configurations used for our neural models

Models	English-Foreign					Foreign-English				
	AER	F1	PRE	REC	ACC	AER	F1	PRE	REC	ACC
`IBM-1 Giza++`	39.03	58.76	59.1	58.43	96.4	42.66	55.39	57.02	53.84	96.2
`IBM-1+NN`	37.64	60.07	62.98	57.41	96.65	39.22	58.53	62.22	55.25	96.57
`IBM-1+NNChar`	36.22	61.55	62.76	60.39	96.69	40.88	56.99	59.75	54.48	96.4
`IBM-1+BPE+B`	31.36	66.52	73.38	60.83	97.32	34.46	63.34	64.35	62.36	96.84
`IBM-1+BPE+B+C`	**31.02**	67.29	72.93	62.45	97.34	**33.93**	63.88	64.99	62.81	96.89
`Fastalign`	28.98	68.75	71.11	66.54	97.35	31.28	66.47	70.73	62.69	97.23
`HMM Giza++`	23.92	73.3	79.23	68.2	97.82	26.33	71.04	79.47	64.23	97.7
`HMM+NN`	26.78	70.95	73.94	68.2	97.55	29.44	68.21	74.69	62.76	97.44
`HMM+NNCharTgt`	26.04	71.57	75.99	67.64	97.64	28.11	69.48	75.59	64.29	97.52
`HMM+NNCharJB`	23.69	73.38	82.38	66.15	97.9	24.9	72.16	83.36	63.61	97.85
`HMM+BPE+B`	19.61	78.25	85.82	71.92	98.25	20.38	77.38	84.28	71.52	98.17
`HMM+BPE+B+C`	**19.17**	79.19	86.61	72.94	98.32	**20.36**	77.41	84.36	71.52	98.17
`IBM-4 Giza++`	21.46	75.48	85.79	67.39	98.08	23.31	73.63	86.56	64.06	97.99

Table 6.6: Alignment error rate (AER), accuracy (ACC), F-score (F1), precision (PRE) and recall (REC) for English-German

Models	English-Foreign					Foreign-English				
	AER	F1	PRE	REC	ACC	AER	F1	PRE	REC	ACC
`HMM+NN`	49.27	50.73	63.52	42.24	97.33	31.45	68.56	67.75	69.39	97.93
`HMM+NNCharTgt`	47.52	52.49	67.71	42.86	97.47	30.94	69.06	75.69	63.51	98.15
`HMM+NNCharJB`	**43.28**	56.73	84.49	42.7	97.88	27.59	72.42	72.7	72.14	98.21
`HMM+BPE+B`	47.03	52.97	62.75	45.83	97.35	27.76	72.24	74.13	70.45	98.24
`HMM+BPE+B+C`	45.85	54.15	64.41	46.71	97.42	**26.05**	73.95	75.61	72.37	98.34

Table 6.7: Alignment error rate (AER), accuracy (ACC), F-score (F1), precision (PRE) and recall (REC) for English-Vietnamese

6.8 Summary

We discussed the benefits and the limitations of using short and long units generated by different BPE configurations. We saw that BPE-based word alignment encourages models to generate more correct one-to-many/many-to-many links, yielding a better recall (Section 6.3). Another benefit of decomposing a word into a sequence of smaller units is that BPE-based models help to get rid of the problem of rare/unknown words (Section 6.5). We also noticed that shorter BPE units mostly change the distribution of many-to-one, one-to-many and many-to-many links (Section 6.4). One drawback of this approach is that if BPE units are too short, length differences between word-based sequences and BPE-based sequences can be large. When this is the case, the alignment task is much more difficult (Section 6.2). We also see that controlling differences between source and target sentence lengths can be a strategy for choosing the right segmentation (e.g., minimizing the average difference in sequence length).

We clearly see the benefits of using GDF after transforming alignment links from BPE level to word level (Section 6.6).

We summarize our findings for selecting a proper BPE configuration for each language pair based on our experiments with `Fastalign`.

- English vs French, German and Czech: These morphologically rich languages do not benefits from too short BPE units, hence their preferred vocabulary size should be in the order of 32K. Note that this is a big reduction for German (see Table 3.2). English can have a smaller vocabulary size such as 4K or 16K. This suggests that too short units for these morphologically rich languages may blur important information regarding words.

- The benefit of using BPE units is less clear for English-Romanian. The small vocabulary size pair 16K-8K only improves over the word-based `Fastalign` in the direction English-Romanian.

- Japanese and Vietnamese benefit most from short BPE units. We recommend an aggressive segmentation into short BPE units, our best results being obtained for 4K for Japanese and 2K for Vietnamese.

These recommended configurations prove their usefulness for our neural models with a gain of AER and F-score (Section 6.7).

Chapter 7

Conclusion

In this closing chapter, we recall the motivations of our work and summarize our contributions. We also identify the main directions for future work.

7.1 Summary

Chapter 1 showed our main motivation: we need neural models that overcome pitfalls of statistical word alignment tools namely `Giza++` and `Fastalign`. Several weaknesses are low-frequency words, no context information in alignment and asymmetrical alignments, etc. In order to comprehensively observe them, a collection of statistical tools is required.

Chapter 2 presented an overview of the alignment task. We defined the alignment problem at various levels from document-level to subword-level. We discussed the most outstanding and recent models in document alignment, sentence alignment and also sub-sentential alignment. With respect to sub-sentential alignment, we mainly presented word alignment models under unsupervised learning and supervised learning. For this alignment level, different types of alignment were introduced and we showed several methods to encode units for word alignment. We also presented the models for phrase alignment and for structure alignment.

Chapter 3 described methods aimed at efficiently evaluate alignment models. We described our training and test corpora for six language pairs English with French, German, Romanian, Czech, Japanese and Vietnamese. We demonstrated that the human reference alignments (sure/possible links) caused bias for the AER metric, a common method to measure model performance. Therefore, we explored a list of methods based on these corpora: analysis about aligned/unaligned words, rare/unknown words, function/content words, word orders, levels of agreement, symmetrization and sentence lengths.

We demonstrated that the baselines do not well predict alignment links for the long sentences. For unaligned words, distortion models of `HMM` and `IBM-4` implemented in `Giza++` do not help to generate more correct links but simply remove incorrect links, creating a large number of incorrectly unaligned words. `HMM Giza++` still has a problem of predicting correctly jumps because of the simple assumptions and the lack of context information. These statistical models also suffer from another problem for rare words called the garbage collector, when rare words in the target language to be misaligned to many source words. In addition, function words are incorrectly aligned to the NULL token. We highlighted that symmetrical alignments and controlling agreement levels are always important approaches to improve our baselines.

Chapter 4 described an overview of artificial neural networks and their applications in NLP. Several common neural network architectures were surveyed: feed-forward neural networks, convolutional neural networks and (bidirectional) recurrent neural networks with long short-term memory. We discussed the three different lines of research: the probabilistic approach, the non-probabilistic approach and the attention-based approach.

Our work belongs to the probabilistic approach where we replace the traditional count-based translation models with several neural network variants, notably contextual models and character-based models. We also neuralized the distortion models using character-based representations. The benefits and limitations of these neural models were shown and discussed compared with `Giza++` and `Fastalign`. One important observation is that neural models can help to achieve remarkable improvements in AER and F-score for most languages pairs, with the higher gains observed for the morphologically rich languages in a small data condition. They also proved their usefulness for rare/unknown words, content words and for long sentences. We noticed that most of these gains are due to a decrease in non-null link errors. In addition, we demonstrated that using a larger training corpus helps to gain more performance points in the case of German.

For neural models, using context helps to disambiguate alignment links for English words by improving the translation distribution. Models using character-based yield significant and consistent gains, especially in small data conditions. They help to differentiate the translation model for rare/unknown words.

Chapter 5 revisited the proposal of Rios et al. [2018] and explored variants of the variational autoencoder models for the unsupervised estimation of neural word alignment models. We underline two promising aspects: (a) using a full model of the joint distribution helps to easily and naturally introduce symmetrization constraints as we showed by proposing two such extensions (Sharing parameters and adding the extra costs rewarding agreement between asymmetric alignments) (b) incorporating monolingual data during training, which especially proves useful in low-resource scenarios.

We see that these techniques can yield competitive results as compared to `Giza++` and to a strong neural network alignment system. Note that the gain is more significant when the morphologically rich language (e.g. Romanian, Czech, German) is on the target side where the emission model is the weakest and benefits most from parameter sharing. Moreover, higher levels of the agreement created by our variants yield better scores in terms of intersection AER.

Chapter 6 presented how to perform the word alignment task by using alignment links between subwords. We explored how different BPE configurations affect word alignment performance. AER, F-score, recall and precision are reported and highlighted the issues of rare words, alignment types, sequence lengths and symmetrization for BPE-based word alignment. In fact, we confirmed that decomposition of a word to a sequence of smaller units get rid of the problem of rare/unknown words. Shorter BPE units encourage different alignment types especially many-to-many links. Moreover, too short BPE units can hurt word-based alignment performance. We finally make recommendation for selecting proper BPE configurations for our six language pairs. French, German and Czech can have a BPE-based vocabulary size 32K, which is much smaller than their word-based vocabulary size. Romanian, Japanese and Vietnamese BPE-based vocabulary size can be smaller e.g., 4K/8K. English can have a vocabulary size such as 4K or 16K.

7.2 Future work

Prediction of unaligned words In our model implementations, unaligned words are paired with a NULL symbol that is simply one special word in the vocabulary, which does not include information of the word that it replaces (The model `CtxCc` encodes context information but fails to bring better performance). In the variational approach, the prediction of null links is quite problematic for the reconstruction component. We showed that our models are strongly inclined to under-generate alignment links, which is detrimental to the overall AER performance. We can see this serious problem in an example of the alignment links generated by one of our best models `HMM+NN+CharJB` in Figure 7.1. Symmetrization (e.g., our variational models) is the first answer to this problem, which however only partly fixes the issue. We highlighted a need for a proper model for the latent representation of the NULL token.

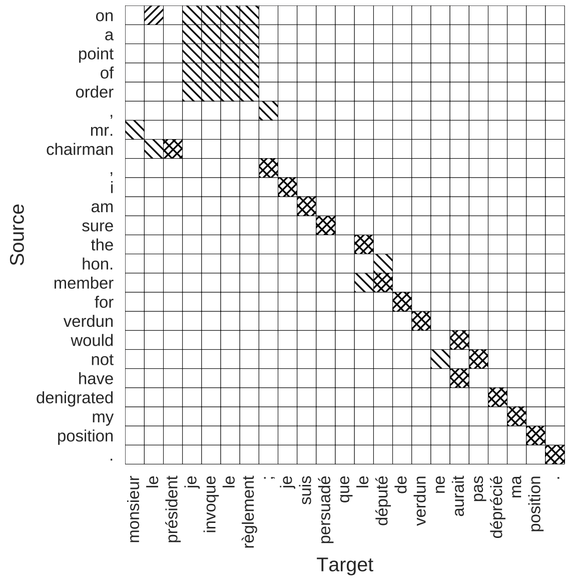


Figure 7.1: Example of the alignment links generated by one of our best models `HMM+NN+CharJB`. Back diagonal hatching, diagonal hatching and crossed diagonal hatching represent a reference alignment link, a predicted alignment link and a correctly predicted alignment link. The phrase “a point of order" is incorrectly aligned to NULL token.

Word orders Our best models having a tendency to concentrate the link distribution around short jumps, a likely sign of a too confident translation model (especially for European languages). Using our neuralized distortions does not seem to fix this issue. We can see this limitation of our models in Figure 7.2. This suggests that much remains to be done in terms of better modeling the distortion.

Many-to-many links Our alignment models are asymmetrical, which limits us to generate more natural alignments. Using subword-level alignment links and then transforming them into word-level alignment links, this approach is always a must-do to obtain more symmetrical alignments. However, our BPE-based models seem to under-generate these links, which suggests two directions of research: (a) a distortion model recognizes word boundaries for subword alignment task, (b) a better technique of transformation from subword-level to word-level alignment.

Optimization problem Another direction of research for our variational models is controlling the optimization problem, a difficult task when their objective functions combine multiple terms with varying dynamics. More work is needed there to design better optimization strategies, with a better balance between the various sub-objectives.

More symmetrical alignment Our mission of finding a symmetrical alignment model is not finished. Sharing decoder parameters and enforcing agreement are a first advance to obtain a more symmetrical alignment model. An approach that we should consider is to enforce the two encoders for source and target sentence to share more information. One possible solution is multilingual encoder that allows to learn both source and target languages.

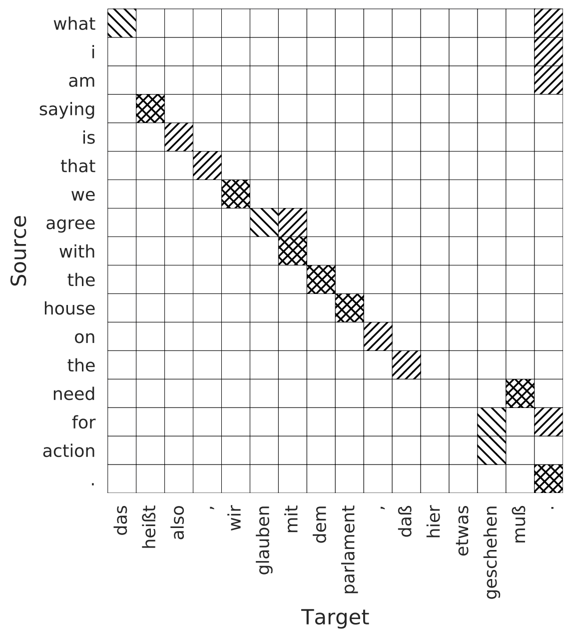


Figure 7.2: Example of the alignment links generated by one of our best models `HMM+NN+CharJB`. Back diagonal hatching, diagonal hatching and crossed diagonal hatching represent a reference alignment link, a predicted alignment link and a correctly predicted alignment link. “is" and “that" are unaligned words. However, for our model, they align with the two German words because our model over-generate jumps of length 1.

Hierarchical syntactic alignment Another area is to develop a neural model, based on structure alignment (Section 2.2.3.3), that predicts production rules such as merging two consecutive word sequences in a monotone order, merging in an inverted order, and aligning source, target words or an empty word. In addition, Corro and Titov [2019] proposed a VAE using the dependency structure of a sentence as a latent variable. We can replace this structure by a more complex form showing the relation between words of source and target sentence (e.g., ITG). This approaches can also yield symmetrical alignments.

7.3 Final words

We presented in this dissertation an overview of alignment tasks and concentrate to explore alignments at word-level and subword-level. We proposed several neural network architectures that are useful for this task. Our alignment neural models establish the strong baselines that more recent models should outperform (Table 7.1). In fact, this research confirms several benefits of using neural networks:

- Neural networks (especially character-based models) improve word representations, yielding a significant gain in alignment accuracy. This overcomes the problem of rare and unknown words in low-resource scenario.

- Fully generative models based on variational autoencoders allow monolingual corpora to improve alignment performance.

- These models permit easily and naturally introducing symmetrization constraints.

We also give the indications to perform the alignment task for the language pairs English with French, German, Romanian, Czech, Japanese and Vietnamese. In addition, we confirm the benefit of using BPE tokenization for this task. We expect that our proposed models and our findings in this dissertation are helpful references for future research.

Models	English-Foreign			Foreign-English		
	Model	AER	F1	Model	AER	F1
English-French	`NNCharJT`	8.41	44.71	`NNCharJT`	7.70	44.45
English-German	`BPE+VAE+SP+AC`	19.13	78.38	`BPE+B+C`	20.36	77.41
English-Romanian	`NNCharWord`	25.51	74.51	`NNCharTgt`	28.01	72.01
English-Czech	`NNCharJT`	15.94	68.31	`BPE+B+C`	17.81	69.09
English-Japanese	`BPE+B+C`	38.3	61.7	`NNCharJB`	37.71	62.29
English-Vietnamese	`NNCharJB`	43.28	56.73	`BPE+B+C`	26.05	73.95

Table 7.1: Our best AER score for each language pair and for each direction. The models `NNChar`, `BPE+VAE`, `BPE+B+C` are respectively described in Section 4.2, Section 5.2 and Section 6.7.

Summary in French

Le **chapitre 1** montrait notre principale motivation: création des modèles neuronaux permettant de résoudre les pièges des modèles d'alignement statistique qui sont par exemple `Giza++` et `Fastalign`. Les différentes faiblesses de ces modèles sont des mots rares, absence d'information contextuelle dans l'alignement et des alignements asymétriques, etc. Afin de les étudier de manière exhaustive, une collection d'outils statistiques est nécessaire.

Le **chapitre 2** présentait la tâche d'alignement. Nous définissions ainsi le problème d'alignement à différents niveaux, du niveau du document au niveau du sous-mot. Nous discutions des modèles les plus remarquables et les plus récents en matière d'alignement de documents, de phrases et de sous-phrases. Concernant l'alignement de sous-phrases, nous présentions principalement des modèles d'alignement de mots en utilisant l'apprentissage non-supervisé et l'apprentissage supervisé. Pour ce niveau d'alignement, les différents types d'alignement étaient introduits et plusieurs méthodes codant les unités pour l'alignement de mots étaient démontrées. Enfin, nous présentions les modèles pour l'alignement de groupes de mots et de structures linguistiques.

Le **chapitre 3** décrivait des méthodes visant à évaluer efficacement les modèles d'alignement. Nous décrivions nos corpus d'entraînement et de test pour six paires de langues composées de l'anglais avec le français, l'allemand, le roumain, le tchèque, le japonais et le vietnamien. Nous démontrions que les alignements de référence humains (liens sûrs/liens possibles) provoquaient un biais lors d'utilisation de la méthode "AER" qui est une méthode connue pour mesurer les performances du modèle. Par conséquent, nous proposions une liste d'autres méthodes basées sur ces corpus : analyse des mots alignés/non-alignés, des mots rares/inconnus, des mots de fonction/contenu, de l'ordre des mots, des niveaux d'accord, de la symétrisation et de la longueur des phrases. Nous démontrions que les modèles référentiels ne prédisent pas correctement les liens d'alignement pour des longues phrases. Pour les mots non alignés, les modèles de distorsion de `HMM` et `IBM-4` implémentés dans `Giza++` n'aident pas à rédiger des liens corrects mais suppriment simplement des liens incorrects, ce qui crée alors un grand nombre de mots incorrectement non-alignés. Le `HMM Giza++` contient un problème de prédiction incorrecte des sauts en raison de la simplicité des hypothèses et du manque d'informations contextuelles. Ces modèles statistiques souffrent également d'un autre problème pour les mots rares appelé le ramasse-miettes, lorsque des mots rares de la langue cible sont mal alignés avec de nombreux mots sources. En outre, les mots de fonction ne sont pas correctement alignés sur le NULL. Nous soulignions que les alignements symétriques et le contrôle des niveaux d'acceptabilité sont toujours des approches importantes pour améliorer ces modèles référentiels.

Le **chapitre 4** décrivait un aperçu général des réseaux de neurones artificiels et de leurs applications en traitement automatique des langues naturelles. Plusieurs architectures communes de réseaux de neurones étaient étudiées : les réseaux de neurones à propagation avant, les réseaux de neurones convolutifs et les réseaux de neurones récurrents (bidirectionnels) avec une mémoire à long terme et à court terme. Nous discutions des trois différents axes de recherche: l'approche probabiliste, l'approche non probabiliste et l'approche axée sur l'attention.

Notre travail s'inscrivait dans l'approche probabiliste où nous remplaçons les modèles de traduction traditionnels par plusieurs variantes de modèles de réseaux neuronaux, notamment des modèles contextuels et des modèles basés sur des caractéres. Nous établissions ainsi les modèles neuronaux de distorsion en utilisant des représentations basées sur des caractères. Nous discutions des avantages et des limites de ces modèles neuronaux par rapport à ceux des

`Giza++` et `Fastalign`. Nous observions que ces modèles neuronaux pourraient contribuer à obtenir des améliorations remarquables de l'ARE et du F-score pour la plupart des paires de langues, notamment des gains en faveur des langues morphologiquement riches dans une réserve de données limitées. Grâce à ces modèles, nous prouvions également leur avantage pour des mots rares/inconnus, des mots de contenu et des phrases longues. Nous remarquions que la majorité de ces points positifs sont dus à une diminution des erreurs de liens non-nulles. Par ailleurs, nous démontrions que l'utilisation d'un corpus de l'entraînement plus large permettrait de gagner des meilleurs points de performance dans le cas de l'allemand.

Dans nos modèles neuronaux, l'utilisation du contexte permettait de lever l'ambiguïté des liens d'alignement des mots anglais en améliorant la distribution de la traduction. Les modèles utilisant des caractères généraient des gains significatifs et cohérents, en particulier dans des réserves de données limitées. Ils aidaient à différencier le modèle de traduction pour des mots rares/inconnus.

Le **chapitre 5** revisitait la proposition de Rios et al. [2018] et explorait des variantes des modèles d'auto-encodeur variationnel pour l'estimation non-supervisée des modèles neuronaux d'alignement de mots. Nous soulignions deux aspects prometteurs: (a) l'utilisation d'un modèle complet de la distribution conjointe permet d'introduire facilement et naturellement des contraintes de symétrisation, comme nous l'avions montré en proposant deux extensions de ce type (partager les paramètres et ajouter les coûts supplémentaires récompensant l'accord entre alignements asymétriques) (b) intégrer des données monolingues pendant l'entraînement, ce qui s'avère particulièrement utile dans les scénarios à faibles ressources.

Nous remarquions que nos techniques peuvent donner des résultats compétitifs par rapport à ceux du `Giza++` et à ceux du système neuronal puissant d'alignement. A noter que le gain est plus significatif en faveur de la langue morphologiquement riche (par exemple le roumain, le tchèque, l'allemand) qui se trouve dans le côté langue cible où le modèle d'émission est le plus faible et profite le plus du partage des paramètres. De plus, des niveaux d'acceptabilité plus élevés crées par nos variantes donnaient de meilleurs scores en termes d'intersection AER.

Le **chapitre 6** présentait comment effectuer la tâche d'alignement de mots en utilisant des liens d'alignement entre des sous-mots. Nous explorions comment différentes configurations BPE affectent les performances d'alignement de mots. L'ARE, le F-score, le rappel et la précision étaient rapportés et soulignaient les problèmes de mots rares, de types d'alignement, de longueurs de séquence et de symétrisation pour l'alignement de mots basé sur BPE. En effet, nous confirmions que la décomposition d'un mot en une séquence d'unités plus petites permet d'éliminer le problème des mots rares/inconnus. Les unités BPE plus courtes encouragent différents types d'alignement, en particulier les liens "many-to-many". De plus, des unités BPE trop courtes peuvent nuire aux performances d'alignement basé sur les mots. Finalement, nous recommandons les configurations BPE appropriées pour nos six paires de langues. Le français, l'allemand et le tchèque peuvent avoir une taille de vocabulaire basée sur BPE 32K, ce qui est beaucoup plus petite que la taille de leur vocabulaire basé sur des mots. La taille du vocabulaire basé sur le BPE roumain, japonais et vietnamien peut être plus petite, par exemple 4K/8K. L'anglais peut avoir une taille de vocabulaire telle que 4K ou 16K.

Bibliography

Ossama Abdel-Hamid, Abdel-Rahman Mohamed, Hui Jiang, Li Deng, Gerald Penn, and Dong Yu. Convolutional neural networks for speech recognition. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 22(10):1533–1545, October 2014. ISSN 2329-9290. doi: 10.1109/TASLP.2014.2339736. URL https://doi.org/10.1109/TASLP.2014.2339736.

Bghiel Afrae, Ben Ahmed Mohamed, and A. Anouar Boudhir. A question answering system with a sequence to sequence grammatical correction. In *Proceedings of the 3rd International Conference on Networking, Information Systems Security*, NISS2020, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450376341. doi: 10.1145/3386723.3387894. URL https://doi.org/10.1145/3386723.3387894.

Lars Ahrenberg. LinES: An English-Swedish parallel treebank. In *Proceedings of the 16th Nordic Conference of Computational Linguistics (NODALIDA 2007)*, pages 270–273, Tartu, Estonia, May 2007. University of Tartu, Estonia. URL https://www.aclweb.org/anthology/W07-2441.

Tamer Alkhouli and Hermann Ney. Biasing attention-based recurrent neural networks using external alignment information. In *Proceedings of the Second Conference on Machine Translation*, pages 108–117, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W17-4711.

Tamer Alkhouli, Gabriel Bretschner, Jan-Thorsten Peter, Mohammed Hethnawi, Andreas Guta, and Hermann Ney. Alignment-based neural machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers*, pages 54–65, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/W16-2206. URL https://www.aclweb.org/anthology/W16-2206.

David Alvarez-Melis and Tommi Jaakkola. A causal framework for explaining the predictions of black-box sequence-to-sequence models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 412–421, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1042. URL https://www.aclweb.org/anthology/D17-1042.

Mikel Artetxe and Holger Schwenk. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Trans. Assoc. Comput. Linguistics*, 7:597–610, 2019. URL https://transacl.org/ojs/index.php/tacl/article/view/1742.

Michael Auli, Michel Galley, Chris Quirk, and Geoffrey Zweig. Joint language and translation modeling with recurrent neural networks. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1044–1054, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D13-1106.

Necip Fazil Ayan and Bonnie J. Dorr. A maximum entropy approach to combining word alignments. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 96–103, New York City, USA, June 2006. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/N06-1013.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL `http://arxiv.org/abs/1409.0473`.

Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/W05-0909`.

Colin Bannard and Chris Callison-Burch. Paraphrasing with bilingual parallel corpora. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 597–604, USA, 2005. Association for Computational Linguistics. doi: 10.3115/1219840.1219914. URL `https://doi.org/10.3115/1219840.1219914`.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3(null):1137–1155, March 2003. ISSN 1532-4435. URL `https://dl.acm.org/doi/10.5555/944919.944966`.

Taylor Berg-Kirkpatrick, Alexandre Bouchard-Côté, John DeNero, and Dan Klein. Painless unsupervised learning with features. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 582–590, Los Angeles, California, June 2010. URL `http://www.aclweb.org/anthology/N10-1083`.

Phil Blunsom and Trevor Cohn. Discriminative word alignment with conditional random fields. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 65–72, Sydney, Australia, July 2006. Association for Computational Linguistics. doi: 10.3115/1220175.1220184. URL `https://www.aclweb.org/anthology/P06-1009`.

Dasha Bogdanova, Cícero dos Santos, Luciano Barbosa, and Bianca Zadrozny. Detecting semantically equivalent questions in online user forums. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 123–131, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.18653/v1/K15-1013. URL `https://www.aclweb.org/anthology/K15-1013`.

Nikolay Bogoychev and Hieu Hoang. Fast and highly parallelizable phrase table for statistical machine translation. In *Proceedings of the First Conference on Machine Translation*, pages 102–109, Berlin, Germany, August 2016. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/W/W16/W16-2211`.

Kaj Bostrom and Greg Durrett. Byte pair encoding is suboptimal for language model pretraining. *CoRR*, abs/2004.03720, 2020. URL `https://arxiv.org/abs/2004.03720`.

Julien Bourdaillet, Stéphane Huet, Fabrizio Gotti, Guy Lapalme, and Philippe Langlais. Enhancing the Bilingual Concordancer TransSearch with Word-Level Alignment. In Yong Gao and Nathalie Japkowicz, editors, *22nd Conference of the Canadian Society for Computational Studies of Intelligence (Canadian AI 2009)*, volume 5549 of *Advances in Artificial Intelligence*, pages 27–38, Kelowna, Canada, May 2009. Springer. doi: 10.1007/978-3-642-01818-3_6. URL `https://hal.archives-ouvertes.fr/hal-02021384`.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. In *Proc. CoNLL*, Berlin, Germany, August 2016. doi: 10.18653/v1/K16-1002. URL `https://www.aclweb.org/anthology/K16-1002`.

Martin Braschler and Peter Schäuble. Multilingual information retrieval based on document alignment techniques. In *Proceedings of the Second European Conference on Research and Advanced Technology for Digital Libraries*, ECDL '98, pages 183–197, Berlin, Heidelberg, 1998. Springer-Verlag. ISBN 3540651012.

Fabienne Braune and Alexander Fraser. Improved unsupervised sentence alignment for symmetrical and asymmetrical parallel corpora. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, COLING '10, pages 81–89, USA, 2010. Association for Computational Linguistics.

Peter F. Brown, Jennifer C. Lai, and Robert L. Mercer. Aligning sentences in parallel corpora. In *29th Annual Meeting of the Association for Computational Linguistics*, pages 169–176, Berkeley, California, USA, June 1991. Association for Computational Linguistics. doi: 10.3115/981344.981366. URL https://www.aclweb.org/anthology/P91-1022.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, Meredith J. Goldsmith, Jan Hajic, Robert L. Mercer, and Surya Mohanty. But dictionaries are data too. In *Proceedings of the Workshop on Human Language Technology:*, Plainsboro, New Jersey, 1993a. URL https://www.aclweb.org/anthology/H93-1039.

Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Comput. Linguist.*, 19(2):263–311, June 1993b. ISSN 0891-2017. URL http://dl.acm.org/citation.cfm?id=972470.972474.

Jamie Brunning, Adria de Gispert, and William Byrne. Context-dependent alignment models for statistical machine translation. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 110–118, Boulder, Colorado, June 2009. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/N/N09/N09-1013.

Matthias Buch-Kromann. Computing translation units and quantifying parallelism in parallel dependency treebanks. In *Proceedings of the Linguistic Annotation Workshop*, pages 69–76, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W07-1512.

Franck Burlot and François Yvon. Learning Morphological Normalization for Translation from and into Morphologically Rich Languages. *The Prague Bulletin of Mathematical Linguistics*, 108:49–60, 2017. doi: 10.1515/pralin-2017-0008. URL https://hal.archives-ouvertes.fr/hal-01618382.

Stanley F. Chen. Aligning sentences in bilingual corpora using lexical information. In *31st Annual Meeting of the Association for Computational Linguistics*, pages 9–16, Columbus, Ohio, USA, June 1993. Association for Computational Linguistics. doi: 10.3115/981574.981576. URL https://www.aclweb.org/anthology/P93-1002.

Wenhu Chen, Evgeny Matusov, Shahram Khadivi, and Jan-Thorsten Peter. Guided alignment training for topic-aware neural machine translation. *CoRR*, abs/1607.01628, 2016. URL http://arxiv.org/abs/1607.01628.

Yong Cheng, Shiqi Shen, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. Agreement-based joint training for bidirectional attention-based neural machine translation. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, pages 2761–2767, New York, New York, USA, 2016. AAAI Press. ISBN 978-1-57735-770-4. URL http://dl.acm.org/citation.cfm?id=3060832.3061007.

Colin Cherry and Dekang Lin. A comparison of syntactically motivated word alignment spaces. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, Trento, Italy, April 2006. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/E06-1019`.

Colin Cherry, George F. Foster, Ankur Bapna, Orhan Firat, and Wolfgang Macherey. Revisiting character-based neural machine translation with capacity and compression. *CoRR*, abs/1808.09943, 2018. URL `http://arxiv.org/abs/1808.09943`.

Kyunghyun Cho. *Foundations and Advances in Deep Learning*. Aalto University publication series DOCTORAL DISSERTATIONS; 21/2014. Aalto University; Aalto-yliopisto, 2014. ISBN 978-952-60-5575-6 (electronic); 978-952-60-5574-9 (printed). URL `http://urn.fi/URN:ISBN:978-952-60-5575-6`.

Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar, October 2014a. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078, 2014b. URL `http://arxiv.org/abs/1406.1078`.

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

Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C. Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 2980–2988, 2015. URL `http://papers.nips.cc/paper/5653-a-recurrent-latent-variable-model-for-sequential-data`.

Junyoung Chung, Kyunghyun Cho, and Yoshua Bengio. A character-level decoder without explicit segmentation for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1693–1703, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1160. URL `https://www.aclweb.org/anthology/P16-1160`.

Martin Čmejrek, Jan Cuřín, Jiří Havelka, Jan Hajič, and Vladislav Kuboň. Prague Czech-English Dependency Treebank. syntactically annotated resources for machine translation. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal, May 2004. European Language Resources Association (ELRA). URL `http://www.lrec-conf.org/proceedings/lrec2004/pdf/745.pdf`.

Trevor Cohn, Cong Duy Vu Hoang, Ekaterina Vymolova, Kaisheng Yao, Chris Dyer, and Gholamreza Haffari. Incorporating structural alignment biases into an attentional neural translation model. *CoRR*, abs/1601.01085, 2016. URL `http://arxiv.org/abs/1601.01085`.

Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 160–167, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781605582054. doi: 10.1145/1390156.1390177. URL `https://doi.org/10.1145/1390156.1390177`.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12: 2493–2537, November 2011. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=1953048.2078186.

Caio Corro and Ivan Titov. Differentiable perturb-and-parse: Semi-supervised parsing with a structured variational autoencoder. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=BJlgNh0qKQ.

Marta R. Costa-jussà and José A. R. Fonollosa. Character-based neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 357–361, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-2058. URL https://www.aclweb.org/anthology/P16-2058.

Fabien Cromières and Sadao Kurohashi. An alignment algorithm using belief propagation and a structure-based distortion model. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 166–174, Athens, Greece, March 2009. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/E09-1020.

Fabien Cromières. *Vers un plus grand lien entre alignement, segmentation et structure des phrases*. PhD thesis, 2010. URL http://www.theses.fr/2010GRENM001. Thèse de doctorat dirigée par Boitet, Christian et Lepage, Yves Informatique Grenoble 2010.

Hoang Cuong and Khalil Sima'an. Latent domain phrase-based models for adaptation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 566–576, Doha, Qatar, October 2014. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D14-1062.

Haskel B. Curry. The method of steepest descent for non-linear minimization problems. *Quarterly of Applied Mathematics*, 2(3):258–261, 1944. ISSN 0033569X, 15524485. URL http://www.jstor.org/stable/43633461.

Lea Cyrus. Building a resource for studying translation shifts. *CoRR*, abs/cs/0606096, 2006. URL http://arxiv.org/abs/cs/0606096.

Ido Dagan, Kenneth Church, and Willian Gale. Robust bilingual word alignment for machine aided translation. In *Very Large Corpora: Academic and Industrial Perspectives*, 1993. URL https://www.aclweb.org/anthology/W93-0301.

In Dawson and In Phelan. *Language files: Materials for an introduction to language and linguistics*. The Ohio State University Press, 2016. URL https://kb.osu.edu/handle/1811/91418.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.

John DeNero and Klaus Macherey. Model-based aligner combination using dual decomposition. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 420–429, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P11-1043.

John DeNero, Alexandre Bouchard-Côté, and Dan Klein. Sampling alignment structure under a Bayesian translation model. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 314–323, Honolulu, Hawaii, October 2008. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D08-1033.

Y. Deng and W. Byrne. HMM word and phrase alignment for statistical machine translation. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(3):494–507, March 2006. ISSN 1558-7916. doi: 10.1109/TASL.2008.916056. URL `http://mi.eng.cam.ac.uk/~wjb31/ppubs/hltemnlp05wtop.pdf`.

Yuntian Deng, Yoon Kim, Justin Chiu, Demi Guo, and Alexander M. Rush. Latent alignment and variational attention. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, pages 9735–9747, 2018. URL `http://papers.nips.cc/paper/8179-latent-alignment-and-variational-attention`.

Leon Derczynski. Complementarity, f-score, and NLP evaluation. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 261–266, Portorož, Slovenia, May 2016. European Language Resources Association (ELRA). URL `https://www.aclweb.org/anthology/L16-1040`.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 2019. doi: 10.18653/v1/n19-1423. URL `https://doi.org/10.18653/v1/n19-1423`.

Shuoyang Ding, Adithya Renduchintala, and Kevin Duh. A call for prudent choice of subword merge operations in neural machine translation. In *Proceedings of Machine Translation Summit XVII Volume 1: Research Track*, pages 204–213, Dublin, Ireland, August 2019a. European Association for Machine Translation. URL `https://www.aclweb.org/anthology/W19-6620`.

Shuoyang Ding, Hainan Xu, and Philipp Koehn. Saliency-driven word alignment interpretation for neural machine translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 1–12, Florence, Italy, August 2019b. Association for Computational Linguistics. doi: 10.18653/v1/W19-5201. URL `https://www.aclweb.org/anthology/W19-5201`.

Yanzhuo Ding, Yang Liu, Huanbo Luan, and Maosong Sun. Visualizing and understanding neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1150–1159, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1106. URL `https://www.aclweb.org/anthology/P17-1106`.

George Doddington. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the Second International Conference on Human Language Technology Research*, HLT '02, pages 138–145, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc.

Carl Doersch. Tutorial on variational autoencoders. *CoRR*, abs/1606.05908, 2016. URL `http://arxiv.org/abs/1606.05908`.

Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning, 2017. URL `https://arxiv.org/abs/1702.08608`.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.*, 12(null):2121–2159, July 2011. ISSN 1532-4435.

Stefan Daniel Dumitrescu, Tiberiu Boros, and Dan Tufis. RACAI's natural language processing pipeline for Universal Dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 174–181, Vancouver, Canada, August 2017. Association for Computational Linguistics. doi: 10.18653/v1/K17-3018. URL `https://www.aclweb.org/anthology/K17-3018`.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. A simple, fast, and effective reparameterization of ibm model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia, June 2013. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/N13-1073`.

Bryan Eikema and Wilker Aziz. Auto-encoding variational neural machine translation. *CoRR*, abs/1807.10564, 2018. URL `http://arxiv.org/abs/1807.10564`.

Miquel Esplà, Mikel Forcada, Gema Ramírez-Sánchez, and Hieu Hoang. ParaCrawl: Web-scale parallel corpora for the languages of the EU. In *Proceedings of Machine Translation Summit XVII Volume 2: Translator, Project and User Tracks*, pages 118–119, Dublin, Ireland, August 2019. European Association for Machine Translation. URL `https://www.aclweb.org/anthology/W19-6721`.

Miquel Esplà-Gomis, Felipe Sánchez-Martínez, and Mikel L. Forcada. Using word alignments to assist computer-aided translation users by marking which target-side words to change or keep unedited. In Mikel L. Forcada, Heidi Depraetere, and Vincent Vandeghinste, editors, *Proceedings of the 15th Annual conference of the European Association for Machine Translation, EAMT 2011, Leuven, Belgium, May 30-31, 2011*. European Association for Machine Translation, 2011. URL `https://www.aclweb.org/anthology/2011.eamt-1.13/`.

Pravallika Etoori, Manoj Chinnakotla, and Radhika Mamidi. Automatic spelling correction for resource-scarce languages using deep learning. In *Proceedings of ACL 2018, Student Research Workshop*, pages 146–152, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-3021. URL `https://www.aclweb.org/anthology/P18-3021`.

Shi Feng, Shujie Liu, Mu Li, and Ming Zhou. Implicit distortion and fertility models for attention-based encoder-decoder NMT model. *CoRR*, abs/1601.03317, 2016. URL `http://arxiv.org/abs/1601.03317`.

Shi Feng, Eric Wallace, Alvin Grissom II au2, Mohit Iyyer, Pedro Rodriguez, and Jordan Boyd-Graber. Pathologies of neural models make interpretations difficult, 2018. URL `https://arxiv.org/abs/1804.07781`.

Alexander Fraser and Daniel Marcu. Measuring word alignment quality for statistical machine translation. *Comput. Linguist.*, 33(3):293–303, September 2007. ISSN 0891-2017. doi: 10.1162/coli.2007.33.3.293. URL `https://doi.org/10.1162/coli.2007.33.3.293`.

Pascale Fung and Percy Cheung. Mining very-non-parallel corpora: Parallel sentence and lexicon extraction via bootstrapping and e. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 57–63, Barcelona, Spain, July 2004a. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/W04-3208`.

Pascale Fung and Percy Cheung. Multi-level bootstrapping for extracting parallel sentences from a quasi-comparable corpus. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 1051–1057, Geneva, Switzerland, aug 23–aug 27 2004b. COLING. URL `https://www.aclweb.org/anthology/C04-1151`.

Philip Gage. A new algorithm for data compression. *C Users J.*, 12(2):23–38, February 1994. ISSN 0898-9788. URL https://dl.acm.org/doi/10.5555/177910.177914.

William A. Gale and Kenneth W. Church. A program for aligning sentences in bilingual corpora. *Computational Linguistics*, 19(1):75–102, 1993. URL https://www.aclweb.org/anthology/J93-1004.

Kuzman Ganchev, João V. Graça, and Ben Taskar. Better alignments = better translations? In *Proceedings of ACL-08: HLT*, pages 986–993, Columbus, Ohio, June 2008a. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P08-1112.

Kuzman Ganchev, Ben Taskar, and Jo ao Gama. Expectation maximization and posterior constraints. In J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 569–576. Curran Associates, Inc., 2008b. URL http://papers.nips.cc/paper/3170-expectation-maximization-and-posterior-constraints.pdf.

Cristina Garbacea and Qiaozhu Mei. Neural language generation: Formulation, methods, and evaluation. *CoRR*, abs/2007.15780, 2020. URL https://arxiv.org/abs/2007.15780.

Sarthak Garg, Stephan Peitz, Udhyakumar Nallasamy, and Matthias Paulik. Jointly learning to align and translate with transformer models. In *Proc. IJCNLP-EMNLP*, Hong Kong, China, November 2019. doi: 10.18653/v1/D19-1453. URL https://www.aclweb.org/anthology/D19-1453.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. Convolutional sequence to sequence learning. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1243–1252. PMLR, 2017. URL http://proceedings.mlr.press/v70/gehring17a.html.

Alan E. Gelfand and Adrian F.M. Smith. Gibbs sampling for marginal posterior expectations. *Communications in Statistics - Theory and Methods*, 20(5-6):1747–1766, 1991. URL https://www.tandfonline.com/doi/abs/10.1080/03610929108830595.

Ulrich Germann. Yawat: Yet Another Word Alignment Tool. In *Proceedings of the ACL-08: HLT Demo Session*, pages 20–23, Columbus, Ohio, June 2008. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P08-4006.

Felix A. Gers, Jürgen A. Schmidhuber, and Fred A. Cummins. Learning to forget: Continual prediction with lstm. *Neural Comput.*, 12(10):2451–2471, October 2000. ISSN 0899-7667. doi: 10.1162/089976600300015015. URL https://doi.org/10.1162/089976600300015015.

L. Getoor and B. Taskar. *An Introduction to Conditional Random Fields for Relational Learning*, pages 93–127. University of Massachusetts, USA, 2007. URL https://people.cs.umass.edu/~mccallum/papers/crf-tutorial.pdf.

Timur Gilmanov, Olga Scrivner, and Sandra Kübler. SWIFT aligner, a multifunctional tool for parallel corpora: Visualization, word alignment, and (morpho)-syntactic cross-language transfer. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2913–2919, Reykjavik, Iceland, May 2014. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2014/pdf/510_Paper.pdf.

Laurent Girin, Simon Leglaive, Xiaoyu Bie, Julien Diard, Thomas Hueber, and Xavier Alameda-Pineda. Dynamical variational autoencoders: A comprehensive review. *CoRR*, abs/2008.12595, 2020. URL https://arxiv.org/abs/2008.12595.

Pierre Godard. *Unsupervised word discovery for computational language documentation. (Découverte non-supervisée de mots pour outiller la linguistique de terrain)*. PhD thesis, University of Paris-Saclay, France, 2019. URL https://tel.archives-ouvertes.fr/tel-02286425.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

João V. Graça, Kuzman Ganchev, and Ben Taskar. Learning tractable word alignment models with complex constraints. *Computational Linguistics*, 36(3):481–504, 2010. doi: 10.1162/coli_a_00007. URL https://www.aclweb.org/anthology/J10-3007.

Yvette Graham and Josef van Genabith. An open source rule induction tool for transfer-based SMT. *Prague Bull. Math. Linguistics*, 91:37–46, 2009. URL http://ufal.mff.cuni.cz/pbml/91/art-graham.pdf.

Alex Graves. Generating sequences with recurrent neural networks. *CoRR*, abs/1308.0850, 2013. URL http://arxiv.org/abs/1308.0850.

Alex Graves and Navdeep Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, pages II–1764–II–1772. JMLR.org, 2014.

Alex Graves and Jürgen Schmidhuber. 2005 special issue: Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Netw.*, 18(5–6):602–610, June 2005. ISSN 0893-6080. doi: 10.1016/j.neunet.2005.06.042. URL https://doi.org/10.1016/j.neunet.2005.06.042.

Alex Graves, Abdel-rahman Mohamed, and Geoffrey E. Hinton. Speech recognition with deep recurrent neural networks. *CoRR*, abs/1303.5778, 2013. URL http://arxiv.org/abs/1303.5778.

Grishman. Iterative alignment of syntactic structures for a bilingual corpus. In *Natural Language Processing Using Very Large Corpora. Text, Speech and Language Technology*. Springer, Dordrecht, 1999.

Declan Groves, Mary Hearne, and Andy Way. Robust sub-sentential alignment of phrase-structure trees. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 1072–1078, Geneva, Switzerland, aug 23–aug 27 2004. COLING. URL https://www.aclweb.org/anthology/C04-1154.

Ishaan Gulrajani, Kundan Kumar, Faruk Ahmed, Adrien Ali Taïga, Francesco Visin, David Vázquez, and Aaron C. Courville. Pixelvae: A latent variable model for natural images. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=BJKYvt5lg.

D. Gupta and R. Yadav. Annotation guidelines for hindi-english word alignment. In *Asian Language Processing, International Conference on*, pages 293–296, Los Alamitos, CA, USA, dec 2010. IEEE Computer Society. doi: 10.1109/IALP.2010.58. URL https://doi.ieeecomputersociety.org/10.1109/IALP.2010.58.

Shizhu He, Cao Liu, Kang Liu, and Jun Zhao. Generating natural answers by incorporating copying and retrieving mechanisms in sequence-to-sequence learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 199–208, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1019. URL https://www.aclweb.org/anthology/P17-1019.

Xiaodong He. Using word dependent transition models in hmm based word alignment for statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, StatMT '07, pages 80–87, Stroudsburg, PA, USA, 2007. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1626355.1626366.

Xuanli He, Gholamreza Haffari, and Mohammad Norouzi. Dynamic programming encoding for subword segmentation in neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3042–3051, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.275. URL https://www.aclweb.org/anthology/2020.acl-main.275.

Enikö Héja. Dictionary building based on parallel corpora and word alignment. In Anne Dykstra and Tanneke Schoonheim, editors, *Proceedings of the 14th EURALEX International Congress*, pages 341–352, Leeuwarden/Ljouwert, The Netherlands, jul 2010. Fryske Akademy. ISBN 978-90-6273-850-3.

Irina Higgins, L.M, A.P, C.B, X.G, M.B, S. M, and A. L. Beta-VAE: Learning basic visual concepts with a constrained variational framework. In *Proc. ICLR*, Toulon, France, 2017.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8): 1735–1780, November 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL https://doi.org/10.1162/neco.1997.9.8.1735.

Matthias Huck, Simon Riess, and Alexander Fraser. Target-side word segmentation strategies for neural machine translation. In *Proceedings of the Second Conference on Machine Translation*, pages 56–67, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-4706. URL https://www.aclweb.org/anthology/W17-4706.

Rebecca Hwa, Philip Resnik, Amy Weinberg, Clara Cabezas, and Okan Kolak. Bootstrapping parsers via syntactic projection across parallel texts. *Nat. Lang. Eng.*, 11(3):311–325, September 2005. ISSN 1351-3249. doi: 10.1017/S1351324905003840. URL https://doi.org/10.1017/S1351324905003840.

Nitin Indurkhya and Fred J. Damerau. *Handbook of Natural Language Processing*. Chapman & Hall/CRC, 2^{nd} edition, 2010. ISBN 1420085921, 9781420085921.

Abraham Ittycheriah and Salim Roukos. A maximum entropy word aligner for Arabic-English machine translation. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 89–96, Vancouver, British Columbia, Canada, October 2005. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/H05-1012.

N. Jardine and C.J. van Rijsbergen. The use of hierarchic clustering in information retrieval. *Information Storage and Retrieval*, 7(5):217 – 240, 1971. ISSN 0020-0271. doi: https://doi.org/10.1016/0020-0271(71)90051-9. URL http://www.sciencedirect.com/science/article/pii/0020027171900519.

Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. On using very large target vocabulary for neural machine translation. In *Proceedings of the 53^{rd} Annual Meeting of the Association for Computational Linguistics and the 7^{th} International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1–10, Beijing, China, July 2015. Association for Computational Linguistics.

M. Junczys-Dowmunt. A phrase table without phrases: Rank encoding for better phrase table compression. In Mauro Cettolo, Marcello Federico, Lucia Specia, and Andy Way, editors, *Proceedings of th 16th International Conference of the European Association for Machine Translation (EAMT)*, pages 245–252, 2012. URL http://www.mt-archive.info/EAMT-2012-Junczys-Dowmunt.

Hidetaka Kamigaito, Taro Watanabe, Hiroya Takamura, and Manabu Okumura. Unsupervised word alignment using frequency constraint in posterior regularized EM. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 153–158, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1017. URL https://www.aclweb.org/anthology/D14-1017.

Moonyoung Kang, Tim Ng, and Long Nguyen. Mandarin word-character hybrid-input neural network language model. In *INTERSPEECH*. INTERSPEECH 2011 12th Annual Conference of the International Speech Communication Association, 2011. URL https://www.isca-speech.org/archive/interspeech_2011/i11_0625.html.

Andrej Karpathy and Fei-Fei Li. Deep visual-semantic alignments for generating image descriptions. *CoRR*, abs/1412.2306, 2014. URL http://arxiv.org/abs/1412.2306.

Andrej Karpathy, Justin Johnson, and Fei-Fei Li. Visualizing and understanding recurrent networks. *CoRR*, abs/1506.02078, 2015. URL http://arxiv.org/abs/1506.02078.

Martin Kay and Martin Roscheisen. Text-translation alignment. *Computational Linguistics*, 19(1):121–142, 1993. URL https://www.aclweb.org/anthology/J93-1006.

Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1181. URL https://www.aclweb.org/anthology/D14-1181.

Yoon Kim, Yacine Jernite, David A. Sontag, and Alexander M. Rush. Character-aware neural language models. *CoRR*, abs/1508.06615, 2015. URL http://arxiv.org/abs/1508.06615.

Yoon Kim, Carl Denton, Luong Hoang, and Alexander M. Rush. Structured attention networks. *CoRR*, abs/1702.00887, 2017. URL http://arxiv.org/abs/1702.00887.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. URL http://arxiv.org/abs/1412.6980.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *Proc. 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6114.

Dietrich Klakow and Jochen Peters. Testing the correlation of word error rate and perplexity. *Speech Communication*, 38(1):19 – 28, 2002. ISSN 0167-6393. doi: https://doi.org/10.1016/S0167-6393(01)00041-3. URL http://www.sciencedirect.com/science/article/pii/S0167639301000413.

Judith Klavans and Evelyne Tzoukermann. The BICORD system combining lexical information from bilingual corpora and machine readable dictionaries. In *COLING 1990 Volume 3: Papers presented to the 13th International Conference on Computational Linguistics*, 1990. URL https://www.aclweb.org/anthology/C90-3031.

Philipp Koehn. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Conference Proceedings: the tenth Machine Translation Summit*, pages 79–86, Phuket, Thailand, 2005. AAMT, AAMT. URL http://mt-archive.info/MTS-2005-Koehn.pdf.

Philipp Koehn. *Statistical Machine Translation.* Cambridge University Press, New York, NY, USA, 1st edition, 2010. ISBN 0521874157, 9780521874151.

Philipp Koehn and Hieu Hoang. Factored translation models. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 868–876, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/D07-1091`.

Philipp Koehn and Rebecca Knowles. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver, August 2017a. URL `http://www.aclweb.org/anthology/W17-3204`.

Philipp Koehn and Rebecca Knowles. Six challenges for neural machine translation. *CoRR*, abs/1706.03872, 2017b. URL `http://arxiv.org/abs/1706.03872`.

Philipp Koehn, Franz J. Och, and Daniel Marcu. Statistical phrase-based translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 127–133, 2003. URL `https://www.aclweb.org/anthology/N03-1017`.

Philipp Koehn, Amittai Axelrod, Alexandra Birch Mayne, Chris Callison-Burch, Miles Osborne, and David Talbot. Edinburgh system description for the 2005 iwslt speech translation evaluation. In *Proceedings of the International Workshop on Spoken Language Translation*, IWSLT, Pittsburgh, PA, 2005.

Shuhei Kondo, Kevin Duh, and Yuji Matsumoto. Hidden markov tree model for word alignment. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

Ivana Kruijff-Korbayová, Klára Chvátalová, and Oana Postolache. Annotation guidelines for Czech-English word alignment. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy, May 2006. European Language Resources Association (ELRA). URL `http://www.lrec-conf.org/proceedings/lrec2006/pdf/575_pdf.pdf`.

Shaohui Kuang, Junhui Li, António Branco, Weihua Luo, and Deyi Xiong. Attention focusing for neural machine translation by bridging source and target embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1767–1776, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1164. URL `https://www.aclweb.org/anthology/P18-1164`.

Taku Kudo. Subword regularization: Improving neural network translation models with multiple subword candidates. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 66–75. Association for Computational Linguistics, 2018. doi: 10.18653/v1/P18-1007. URL `https://www.aclweb.org/anthology/P18-1007/`.

Taku Kudo and John Richardson. 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*, pages 66–71, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-2012. URL `https://www.aclweb.org/anthology/D18-2012`.

Tz-Liang Kueng and Keh-Yih Su. A robust cross-style bilingual sentences alignment model. In *COLING 2002: The 19th International Conference on Computational Linguistics*, 2002. URL `https://www.aclweb.org/anthology/C02-1009`.

Oi Yee Kwong, Benjamin K. Tsou, Tom B.Y. Lai, Robert W.P. Luk, Lawrence Y.L. Cheung, and Francis C.Y. Chik. Alignment and extraction of bilingual legal terminology from context profiles. In *COLING-02: COMPUTERM 2002: Second International Workshop on Computational Terminology*, 2002. URL https://www.aclweb.org/anthology/W02-1404.

Matthieu Labeau and Alexandre Allauzen. Character and subword-based word representation for neural language modeling prediction. In *Proceedings of the First Workshop on Subword and Character Level Models in NLP*, pages 1–13, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W17-4101.

Patrik Lambert, Adrià De Gispert, Rafael Banchs, and José B. Mariño. Guidelines for word alignment evaluation and manual alignment. *Language Resources and Evaluation*, 39(4):267–285, 2005. ISSN 1574020X, 15728412. URL http://www.jstor.org/stable/30204533.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1030. URL https://www.aclweb.org/anthology/N16-1030.

Guillaume Lample, L.D, and M.A.R. Unsupervised machine translation using monolingual corpora only. *CoRR*, abs/1711.00043, 2017. URL http://arxiv.org/abs/1711.00043.

Fethi Lamraoui and Philippe Langlais. Yet another fast, robust and open source sentence aligner. time to reconsider sentence alignment? In *XIV Machine Translation Summit*, Nice, France, Sept. 2013. URL http://www.iro.umontreal.ca/~felipe/bib2webV0.81/cv/papers/MTSummit-2013-Fethi.pdf.

Wuwei Lan and Wei Xu. Character-based neural networks for sentence pair modeling. In Marilyn A. Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pages 157–163. Association for Computational Linguistics, 2018. doi: 10.18653/v1/n18-2025. URL https://doi.org/10.18653/v1/n18-2025.

Philippe Langlais, George Foster, and Guy Lapalme. TransType: a computer-aided translation typing system. In *ANLP-NAACL 2000 Workshop: Embedded Machine Translation Systems*, 2000. URL https://www.aclweb.org/anthology/W00-0507.

Adrien Lardilleux, Yves Lepage, and François Yvon. The contribution of low frequencies to multilingual sub-sentential alignment: a differential associative approach. *International Journal of Advanced Intelligence*, 3(2):189–217, 2011.

Adrien Lardilleux, François Yvon, and Yves Lepage. Hierarchical Sub-sentential Alignment with Anymalign. In *16th annual conference of the European Association for Machine Translation (EAMT 2012)*, pages 279–286, Trento, Italy, May 2012. URL https://hal.archives-ouvertes.fr/hal-00747385.

Adrien Lardilleux, François Yvon, and Yves Lepage. Generalizing sampling-based multilingual alignment. *Machine Translation*, 27(1):1–23, March 2013. doi: 10.1007/s10590-012-9126-0. URL https://hal.archives-ouvertes.fr/hal-00753859.

Alon Lavie, Alok Parlikar, and Vamshi Ambati. Syntax-driven learning of sub-sentential translation equivalents and translation rules from parsed parallel corpora. In *Proceedings of the*

ACL-08: HLT Second Workshop on Syntax and Structure in Statistical Translation (SSST-2), pages 87–95, Columbus, Ohio, June 2008. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W08-0411.

Benoit Lavoie, Michael White, and Tanya Korelsky. Inducing lexico-structural transfer rules from parsed bi-texts. In *Proceedings of the ACL 2001 Workshop on Data-Driven Methods in Machine Translation*, 2001. URL https://www.aclweb.org/anthology/W01-1403.

Hong Phuong Le, Thi Minh Huyen Nguyen, Azim Roussanaly, and Tuong Vinh Ho. A Hybrid Approach to Word Segmentation of Vietnamese Texts. In *2nd International Conference on Language and Automata Theory and Applications - LATA 2008*, volume 5196 of *Lecture Notes in Computer Science*, pages 240–249, Tarragona, Spain, March 2008. Springer Berlin / Heidelberg. doi: 10.1007/978-3-540-88282-4_23. URL https://hal.inria.fr/inria-00334761. The original publication is available at www.springerlink.com.

Phuong Le-Hong, Azim Roussanaly, Thi Minh Huyen Nguyen, and Mathias Rossignol. An empirical study of maximum entropy approach for part-of-speech tagging of vietnamese texts. In *Actes de la 17e conférence sur le Traitement Automatique des Langues Naturelles*, Montréal, Canada, July 2010. Association pour le Traitement Automatique des Langues. URL http://www.atala.org/taln_archives/TALN/TALN-2010/taln-2010-long-036.

Jason Lee, Kyunghyun Cho, and Thomas Hofmann. Fully character-level neural machine translation without explicit segmentation. *Transactions of the Association for Computational Linguistics*, 5:365–378, 2017. doi: 10.1162/tacl_a_00067. URL https://www.aclweb.org/anthology/Q17-1026.

Joël Legrand, Michael Auli, and Ronan Collobert. Neural network-based word alignment through score aggregation. In *Proceedings of the First Conference on Machine Translation*, pages 66–73, Berlin, Germany, August 2016. Association for Computational Linguistics.

M. Paul Lewis. *Ethnologue: Languages of the World*. SIL International, Dallas, TX, USA, sixteenth edition, 2009.

Jian Li, Zhaopeng Tu, Baosong Yang, Michael R. Lyu, and Tong Zhang. Multi-head attention with disagreement regularization. *CoRR*, abs/1810.10183, 2018a. URL http://arxiv.org/abs/1810.10183.

Jiwei Li, Xinlei Chen, Eduard H. Hovy, and Dan Jurafsky. Visualizing and understanding neural models in NLP. In Kevin Knight, Ani Nenkova, and Owen Rambow, editors, *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 681–691. The Association for Computational Linguistics, 2016. doi: 10.18653/v1/n16-1082. URL https://doi.org/10.18653/v1/n16-1082.

Peng Li, Maosong Sun, and Ping Xue. Fast-champollion: A fast and robust sentence alignment algorithm. In *Coling 2010: Posters*, pages 710–718, Beijing, China, August 2010. Coling 2010 Organizing Committee. URL https://www.aclweb.org/anthology/C10-2081.

Xintong Li, Lemao Liu, Zhaopeng Tu, Shuming Shi, and Max Meng. Target foresight based attention for neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1380–1390, New Orleans, Louisiana, June 2018b. Association for Computational Linguistics. doi: 10.18653/v1/N18-1125. URL https://www.aclweb.org/anthology/N18-1125.

Percy Liang, Ben Taskar, and Dan Klein. Alignment by agreement. In *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, HLT-NAACL '06, pages 104–111, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/W04-1013`.

Dekang Lin. Automatic identification of non-compositional phrases. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 317–324, College Park, Maryland, USA, June 1999. Association for Computational Linguistics. doi: 10.3115/1034678.1034730. URL `https://www.aclweb.org/anthology/P99-1041`.

Chunyang Liu, Yang Liu, Maosong Sun, Huanbo Luan, and Heng Yu. Generalized agreement for bidirectional word alignment. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1828–1836, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1210. URL `https://www.aclweb.org/anthology/D15-1210`.

Lemao Liu, Masao Utiyama, Andrew Finch, and Eiichiro Sumita. Neural machine translation with supervised attention. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3093–3102, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee. URL `https://www.aclweb.org/anthology/C16-1291`.

Shujie Liu, Nan Yang, Mu Li, and Ming Zhou. A recursive recurrent neural network for statistical machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1491–1500, Baltimore, Maryland, June 2014. Association for Computational Linguistics. doi: 10.3115/v1/P14-1140. URL `https://www.aclweb.org/anthology/P14-1140`.

Xiaodong Liu, Kevin Duh, and Yuji Matsumoto. Topic models + word alignment = a flexible framework for extracting bilingual dictionary from comparable corpus. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 212–221, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/W13-3523`.

Yang Liu, Qun Liu, and Shouxun Lin. Log-linear models for word alignment. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 459–466, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. doi: 10.3115/1219840.1219897. URL `https://www.aclweb.org/anthology/P05-1057`.

Zihan Liu, Yan Xu, Genta Indra Winata, and Pascale Fung. Incorporating word and subword units in unsupervised machine translation using language model rescoring. *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, 2019. doi: 10.18653/v1/w19-5327. URL `http://dx.doi.org/10.18653/v1/W19-5327`.

Minh-Thang Luong and Christopher D. Manning. Stanford neural machine translation systems for spoken language domain. In *International Workshop on Spoken Language Translation*, Da Nang, Vietnam, 2015.

Minh-Thang Luong and Christopher D. Manning. Achieving open vocabulary neural machine translation with hybrid word-character models. In *Association for Computational Linguistics (ACL)*, Berlin, Germany, August 2016. URL `https://nlp.stanford.edu/pubs/luong2016acl_hybrid.pdf`.

Thang Luong, Richard Socher, and Christopher Manning. Better word representations with recursive neural networks for morphology. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 104–113, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W13-3512.

Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

Scott M. Lynch. *Introduction to Applied Bayesian Statistics and Estimation for Social Scientists*. Springer Science & Business Media, LLC., New York, NY, April 2007. URL http://faculty.ksu.edu.sa/69424/us_BOOk/Introduction%20to%20Applied%20Bayesian%20Statistics.pdf.

Daniel Marcu and William Wong. A phrase-based, joint probability model for statistical machine translation. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, EMNLP '02, pages 133–139, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1118693.1118711. URL https://doi.org/10.3115/1118693.1118711.

David Mareček. Czech-English manual word alignment. Technical report, LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University, 2016. URL http://hdl.handle.net/11234/1-1804.

Yuji Matsumoto, Takehito Utsuro, and Hiroyuki Ishimoto. Structural matching of parallel texts. In *31st Annual Meeting of the Association for Computational Linguistics*, pages 23–30, Columbus, Ohio, USA, June 1993. Association for Computational Linguistics. doi: 10.3115/981574.981578. URL https://www.aclweb.org/anthology/P93-1004.

Evgeny Matusov, Richard Zens, and Hermann Ney. Symmetric word alignments for statistical machine translation. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 219–225, Geneva, Switzerland, aug 23–aug 27 2004. COLING. URL https://www.aclweb.org/anthology/C04-1032.

Richard T. McCoy and Robert Frank. Phonologically informed edit distance algorithms for word alignment with low-resource languages. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2018*, pages 102–112, 2018. URL http://aclweb.org/anthology/W18-0311.

I. Dan Melamed. Automatic construction of clean broad-coverage translation lexicons. In *Conference of the Association for Machine Translation in the Americas*, Montreal, Canada, October 2-5 1996a. URL https://www.aclweb.org/anthology/1996.amta-1.13.

I. Dan Melamed. A geometric approach to mapping bitext correspondence. In *Conference on Empirical Methods in Natural Language Processing*, 1996b. URL https://www.aclweb.org/anthology/W96-0201.

I. Dan Melamed. Bitext maps and alignment via pattern recognition. *Computational Linguistics*, 25(1):107–130, 1999. URL https://www.aclweb.org/anthology/J99-1003.

I. Dan Melamed. Models of translational equivalence among words. *Computational Linguistics*, 26(2):221–249, 2000. doi: 10.1162/089120100561683. URL https://doi.org/10.1162/089120100561683.

I. Dan Melamed. *Empirical Methods for Exploiting Parallel Texts*. The MIT Press, 01 2001. ISBN 9780262279642. doi: 10.7551/mitpress/2708.001.0001. URL `https://doi.org/10.7551/mitpress/2708.001.0001`.

Jamshed Memon, Maira Sami, Rizwan Ahmed Khan, and Mueen Uddin. Handwritten optical character recognition (OCR): A comprehensive systematic literature review (SLR). *IEEE Access*, 8:142642–142668, 2020. doi: 10.1109/ACCESS.2020.3012542. URL `https://doi.org/10.1109/ACCESS.2020.3012542`.

Arul Menezes and Stephen D. Richardson. A best-first alignment algorithm for automatic extraction of transfer mappings from bilingual corpora. In *Proceedings of the ACL 2001 Workshop on Data-Driven Methods in Machine Translation*. Association for Computational Linguistics, 2001. URL `https://www.aclweb.org/anthology/W01-1406`.

Haitao Mi, Zhiguo Wang, and Abe Ittycheriah. Supervised attentions for neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2283–2288, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1249. URL `https://www.aclweb.org/anthology/D16-1249`.

Yishu Miao, Lei Yu, and Phil Blunsom. Neural variational inference for text processing. In Maria-Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 1727–1736. JMLR.org, 2016. URL `http://proceedings.mlr.press/v48/miao16.html`.

Rada Mihalcea and Ted Pedersen. An evaluation exercise for word alignment. In *Proceedings of the HLT-NAACL 2003 Workshop on Building and Using Parallel Texts: Data Driven Machine Translation and Beyond - Volume 3*, HLT-NAACL-PARALLEL '03, pages 1–10, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. doi: 10.3115/1118905.1118906.

T. Mikolov, S. Kombrink, L. Burget, J. Černocký, and S. Khudanpur. Extensions of recurrent neural network language model. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011.

Tomas Mikolov, Martin Karafiát, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur. Recurrent neural network based language model. In Takao Kobayashi, Keikichi Hirose, and Satoshi Nakamura, editors, *INTERSPEECH*, pages 1045–1048. ISCA, 2010. URL `http://dblp.uni-trier.de/db/conf/interspeech/interspeech2010.html#MikolovKBCK10`.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. *CoRR*, abs/1310.4546, 2013. URL `http://arxiv.org/abs/1310.4546`.

Robert C. Moore. Fast and accurate sentence alignment of bilingual corpora. In *Proceedings of the 5th Conference of the Association for Machine Translation in the Americas on Machine Translation: From Research to Real Users*, AMTA '02, pages 135–144, Berlin, Heidelberg, 2002. Springer-Verlag. ISBN 3540442820.

Robert C. Moore. Improving ibm word-alignment model 1. In *Proceedings of the 42^{Nd} Annual Meeting on Association for Computational Linguistics*, ACL '04, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics. doi: 10.3115/1218955.1219021.

Robert C. Moore. A discriminative framework for bilingual word alignment. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language*

Processing, HLT '05, pages 81–88, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

Makoto Morishita, Yusuke Oda, Graham Neubig, Koichiro Yoshino, Katsuhito Sudoh, and Satoshi Nakamura. An empirical study of mini-batch creation strategies for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 61–68, Vancouver, August 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-3208. URL https://www.aclweb.org/anthology/W17-3208.

Makoto Morishita, Jun Suzuki, and Masaaki Nagata. Improving neural machine translation by incorporating hierarchical subword features. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 618–629, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/C18-1052.

Dragos Stefan Munteanu and Daniel Marcu. *Exploiting Comparable Corpora*, pages 205–222. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. ISBN 978-3-642-20128-8. doi: 10.1007/978-3-642-20128-8_11. URL https://doi.org/10.1007/978-3-642-20128-8_11.

Makoto Nagao. A framework of a mechanical translation between japanese and english by analogy principle. In *Proc. of the International NATO Symposium on Artificial and Human Intelligence*, pages 173–180, USA, 1984. Elsevier North-Holland, Inc. ISBN 0444865454.

Masaaki Nagata, Chousa Katsuki, and Masaaki Nishino. A supervised word alignment method based on cross-language span prediction using multilingual BERT. *CoRR*, abs/2004.14516, 2020. URL https://arxiv.org/abs/2004.14516.

Graham Neubig. The Kyoto free translation task, 2011. URL http://www.phontron.com/kftt.

Ahn-Khoa Ngo-Ho and François Yvon. Neural Baselines for Word Alignments. In *Proc. IWSLT*, Hong-Kong, China, November 2019. URL https://hal.archives-ouvertes.fr/hal-02343217.

Anh Khoa Ngo Ho. Companion report to the PhD dissertation: "Generative Probabilistic Alignment Models for Words and Subwords: a Systematic Exploration of the Limits and Potentials of Neural Parametrizations". Technical report, Université Paris Saclay ; Laboratoire Interdisciplinaire des Sciences du Numérique, February 2021. URL https://hal.archives-ouvertes.fr/hal-03153752.

Anh Khoa Ngo Ho and François Yvon. Generative latent neural models for automatic word alignment. In *Proceedings of the 14th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 64–77, Virtual, October 2020. Association for Machine Translation in the Americas. URL https://www.aclweb.org/anthology/2020.amta-research.6.

Garrett Nicolai, Saeed Najafi, and Grzegorz Kondrak. String transduction with target language models and insertion handling. *CoRR*, abs/1809.07182, 2018. URL http://arxiv.org/abs/1809.07182.

Jan Niehues and Stephan Vogel. Discriminative word alignment via alignment matrix modeling. In *Proceedings of the Third Workshop on Statistical Machine Translation*, StatMT '08, pages 18–25, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics. ISBN 978-1-932432-09-1.

Sonja Nießen and Hermann Ney. Improving smt quality with morpho-syntactic analysis. In *Proceedings of the 18th Conference on Computational Linguistics - Volume 2*, COLING '00, pages 1081–1085, USA, 2000. Association for Computational Linguistics. doi: 10.3115/992730.992809. URL `https://doi.org/10.3115/992730.992809`.

Masaaki Nishino, Jun Suzuki, and Masaaki Nagata. Phrase table pruning via submodular function maximization. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 406–411, Berlin, Germany, August 2016. Association for Computational Linguistics. URL `http://anthology.aclweb.org/P16-2066`.

Chigozie Nwankpa, Winifred Ijomah, Anthony Gachagan, and Stephen Marshall. Activation functions: Comparison of trends in practice and research for deep learning. *CoRR*, abs/1811.03378, 2018. URL `http://arxiv.org/abs/1811.03378`.

Franz Josef Och. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 160–167, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. doi: 10.3115/1075096.1075117.

Franz Josef Och and Hermann Ney. A comparison of alignment models for statistical machine translation. In *Proceedings of the 18th Conference on Computational Linguistics - Volume 2*, COLING '00, pages 1086–1090, Stroudsburg, PA, USA, 2000. Association for Computational Linguistics. doi: 10.3115/992730.992810.

Franz Josef Och and Hermann Ney. A systematic comparison of various statistical alignment models. *Comput. Linguistics*, 29(1):19–51, 2003. doi: 10.1162/089120103321337421. URL `https://doi.org/10.1162/089120103321337421`.

Franz Josef Och and Hermann Ney. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30(4), 2004. URL `http://acl.ldc.upenn.edu/J/J04/J04-4002.pdf`.

Franz Josef Och and Hans Weber. Improving statistical natural language translation with categories and rules. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 2*, pages 985–989, Montreal, Quebec, Canada, August 1998. Association for Computational Linguistics. doi: 10.3115/980691.980731. URL `https://www.aclweb.org/anthology/P98-2162`.

Franz Josef Och, Christoph Tillmann, and Hermann Ney. Improved alignment models for statistical machine translation. In *1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 20–28, 1999. URL `http://www.aclweb.org/anthology/W99-0604`.

James O'Neill and Danushka Bollegala. Semi-supervised multi-task word embeddings. *CoRR*, abs/1809.05886, 2018. URL `http://arxiv.org/abs/1809.05886`.

Robert Östling and Jörg Tiedemann. Efficient word alignment with markov chain monte carlo. *The Prague Bulletin of Mathematical Linguistics*, 106(1):125 – 146, 2016. doi: https://doi.org/10.1515/pralin-2016-0013. URL `https://content.sciendo.com/view/journals/pralin/106/1/article-p125.xml`.

Gustavo Paetzold, Fernando Alva-Manchego, and Lucia Specia. MASSAlign: Alignment and annotation of comparable documents. In *Proceedings of the IJCNLP 2017, System Demonstrations*, pages 1–4, Tapei, Taiwan, November 2017. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/I17-3001`.

Artidoro Pagnoni, Kevin Liu, and Shangyan Li. Conditional variational autoencoder for neural machine translation. *CoRR*, abs/1812.04405, 2018. URL http://arxiv.org/abs/1812.04405.

Santanu Pal, Tanmoy Chakraborty, and Sivaji Bandyopadhyay. Handling multiword expressions in phrase-based statistical machine translation. In *Proceedings of the 13th Machine Translation Summit (MT Summit XIII)*, pages 215–224. International Association for Machine Translation, 2011. URL http://www.mt-archive.info/MTS-2011-Pal.pdf.

Bo Pang, Kevin Knight, and Daniel Marcu. Syntax-based alignment of multiple translations: Extracting paraphrases and generating new sentences. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 181–188, 2003. URL https://www.aclweb.org/anthology/N03-1024.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL https://www.aclweb.org/anthology/P02-1040.

Alexandre Patry and Philippe Langlais. Automatic identification of parallel documents with light or without linguistic resources. In *Proceedings of the 18th Canadian Society Conference on Advances in Artificial Intelligence*, AI'05, pages 354–365, Berlin, Heidelberg, 2005. Springer-Verlag. ISBN 3540258647. doi: 10.1007/11424918_37. URL https://doi.org/10.1007/11424918_37.

Jan-Thorsten Peter, Arne Nix Nix, and Hermann Ney. Generating alignments using target foresight in attention-based neural machine translation. In *Conference of the European Association for Machine Translation*, pages 27–36, Prague, Czech Republic, June 2017.

Minh Quang Pham, Josep Crego, Jean Senellart, and François Yvon. Fixing translation divergences in parallel corpora for neural MT. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2967–2973, Brussels, Belgium, 2018. doi: 10.18653/v1/D18-1328. URL http://aclweb.org/anthology/D18-1328.

Bruno Pouliquen, Ralf Steinberger, Camelia Ignat, Emilia Käsper, and Irina Temnikova. Multilingual and cross-lingual news topic tracking. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 959–965, Geneva, Switzerland, aug 23–aug 27 2004. COLING. URL https://www.aclweb.org/anthology/C04-1138.

David M. W. Powers. Applications and explanations of Zipf's law. In *New Methods in Language Processing and Computational Natural Language Learning*, 1998. URL https://www.aclweb.org/anthology/W98-1218.

Yunchen Pu, Zhe Gan, Ricardo Henao, Xin Yuan, Chunyuan Li, Andrew Stevens, and Lawrence Carin. Variational autoencoder for deep learning of images, labels and captions. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 2352–2360, 2016. URL http://papers.nips.cc/paper/6528-variational-autoencoder-for-deep-learning-of-images-labels-and-captions.

Chris Quirk, Chris Brockett, and William Dolan. Monolingual machine translation for paraphrase generation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 142–149, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W04-3219.

Adithya Renduchintala, Pamela Shapiro, Kevin Duh, and Philipp Koehn. Character-aware decoder for neural machine translation. *CoRR*, abs/1809.02223, 2018. URL http://arxiv.org/abs/1809.02223.

Philip Resnik. A perspective on word sense disambiguation methods and their evaluation. In *Tagging Text with Lexical Semantics: Why, What, and How?*, 1997. URL https://www.aclweb.org/anthology/W97-0213.

Philip Resnik. Mining the web for bilingual text. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 527–534, College Park, Maryland, USA, June 1999. Association for Computational Linguistics. doi: 10.3115/1034678.1034757. URL https://www.aclweb.org/anthology/P99-1068.

Philip Resnik and Noah A. Smith. The web as a parallel corpus. *Computational Linguistics*, 29(3):349–380, 2003. doi: 10.1162/089120103322711578. URL https://www.aclweb.org/anthology/J03-3002.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and variational inference in deep latent gaussian models. In *Proceedings of the 31st International Conference on International Conference on Machine Learning*, volume 2, 2014. URL https://dl.acm.org/doi/10.5555/3044805.3045035.

Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should I trust you?": Explaining the predictions of any classifier. In Balaji Krishnapuram, Mohak Shah, Alexander J. Smola, Charu C. Aggarwal, Dou Shen, and Rajeev Rastogi, editors, *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 1135–1144. ACM, 2016. doi: 10.1145/2939672.2939778. URL https://doi.org/10.1145/2939672.2939778.

Stefan Riezler, Alexander Vasserman, Ioannis Tsochantaridis, Vibhu Mittal, and Yi Liu. Statistical machine translation for query expansion in answer retrieval. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 464–471, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P07-1059.

A Rios, A Göhring, and Martin Volk. A quechua-spanish parallel treebank. In *7th Conference on Treebanks and Linguistic Theories*, 2009. URL https://doi.org/10.5167/uzh-20593.

Miguel Rios, Wilker Aziz, and Khalil Sima'an. Deep generative model for joint alignment and word representation. In Marilyn A. Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 1011–1023. Association for Computational Linguistics, 2018. doi: 10.18653/v1/n18-1092. URL https://doi.org/10.18653/v1/n18-1092.

Tony Robinson, Mike Hochberg, and Steve Renals. *The Use of Recurrent Neural Networks in Continuous Speech Recognition*, pages 233–258. Springer US, Boston, MA, 1996. ISBN 978-1-4613-1367-0. doi: 10.1007/978-1-4613-1367-0_10. URL https://doi.org/10.1007/978-1-4613-1367-0_10.

Sebastian Ruder. An overview of gradient descent optimization algorithms, 2017. URL https://arxiv.org/abs/1609.04747.

Masoud Jalili Sabet, Philipp Dufter, and Hinrich Schütze. Simalign: High quality word alignments without parallel training data using static and contextualized embeddings. *CoRR*, abs/2004.08728, 2020. URL https://arxiv.org/abs/2004.08728.

Markus Saers and Dekai Wu. Improving phrase-based translation via word alignments from Stochastic Inversion Transduction Grammars. In *Proceedings of the Third Workshop on Syntax and Structure in Statistical Translation (SSST-3) at NAACL HLT 2009*, pages 28–36, Boulder, Colorado, June 2009. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/W09-2304`.

Hasim Sak, Andrew W. Senior, and Françoise Beaufays. Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition. *CoRR*, abs/1402.1128, 2014. URL `http://arxiv.org/abs/1402.1128`.

Baskaran Sankaran, Haitao Mi, Yaser Al-Onaizan, and Abe Ittycheriah. Temporal attention model for neural machine translation. *CoRR*, abs/1608.02927, 2016. URL `http://arxiv.org/abs/1608.02927`.

Philip Schulz, Wilker Aziz, and Khalil Sima'an. Word alignment without NULL words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 169–174, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-2028. URL `https://www.aclweb.org/anthology/P16-2028`.

M. Schuster and K.K. Paliwal. Bidirectional recurrent neural networks. *Trans. Sig. Proc.*, 45 (11):2673–2681, November 1997. ISSN 1053-587X. doi: 10.1109/78.650093. URL `https://doi.org/10.1109/78.650093`.

Holger Schwenk. Filtering and mining parallel data in a joint multilingual space. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 228–234, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-2037. URL `https://www.aclweb.org/anthology/P18-2037`.

Abigail See, Minh-Thang Luong, and Christopher D. Manning. Compression of neural machine translation models via pruning. In *Computational Natural Language Learning (CoNLL)*, 2016. URL `https://nlp.stanford.edu/pubs/see2016compression.pdf`.

Rico Sennrich and Martin Volk. Iterative, MT-based sentence alignment of parallel texts. In *Proceedings of the 18th Nordic Conference of Computational Linguistics (NODALIDA 2011)*, pages 175–182, Riga, Latvia, May 2011. Northern European Association for Language Technology (NEALT). URL `https://www.aclweb.org/anthology/W11-4624`.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. *CoRR*, abs/1508.07909, 2015. URL `http://arxiv.org/abs/1508.07909`.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, August 2016. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P16-1162`.

Rico Sennrich, Orhan Firat, Kyunghyun Cho, Alexandra Birch, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel Läubli, Antonio Valerio Miceli Barone, Jozef Mokry, and Maria Nădejde. Nematus: a Toolkit for Neural Machine Translation. *arXiv e-prints*, art. arXiv:1703.04357, Mar 2017.

Pamela Shapiro and Kevin Duh. BPE and charcnns for translation of morphology: A cross-lingual comparison and analysis. *CoRR*, abs/1809.01301, 2018. URL `http://arxiv.org/abs/1809.01301`.

Michel Simard, George F. Foster, and Pierre Isabelle. Using cognates to align sentences in bilingual corpora. In *Proceedings of the 1993 Conference of the Centre for Advanced Studies on Collaborative Research: Distributed Computing - Volume 2*, CASCON '93, pages 1071–1082. IBM Press, 1993.

Andrei Simion, Michael Collins, and Cliff Stein. On a strictly convex IBM model 1. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 221–226, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1023. URL `https://www.aclweb.org/anthology/D15-1023`.

Anil Kumar Singh and Samar Husain. Comparison, selection and use of sentence alignment algorithms for new language pairs. In *Proceedings of the ACL Workshop on Building and Using Parallel Texts*, pages 99–106, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/W05-0816`.

David A. Smith and Noah A. Smith. Bilingual parsing with factored estimation: Using English to parse Korean. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 49–56, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/W04-3207`.

Noah A. Smith and Michael E. Jahr. Cairo: An alignment visualization tool. In *Proceedings of the Second International Conference on Language Resources and Evaluation, LREC 2000, 31 May - June 2, 2000, Athens, Greece*. European Language Resources Association, 2000. URL `http://www.lrec-conf.org/proceedings/lrec2000/html/summary/58.htm`.

David Sontag, Amir Globerson, and Tommi Jaakkola. *Introduction to Dual Decomposition for Inference*. MIT Press, optimization in machine learning edition, January 2010. URL `https://www.microsoft.com/en-us/research/publication/introduction-to-dual-decomposition-for-inference/`.

L. Specia, C. Scarton, G. H. Paetzold, and G. Hirst. *Quality Estimation for Machine Translation*. Morgan & Claypool Publishers, 2018. URL `https://www.morganclaypool.com/doi/10.2200/S00854ED1V01Y201805HLT039`.

Felix Stahlberg, Danielle Saunders, and Bill Byrne. An operation sequence model for explainable neural machine translation. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 175–186, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5420. URL `https://www.aclweb.org/anthology/W18-5420`.

Ralf Steinberger, Bruno Pouliquen, and Johan Hagman. Cross-lingual document similarity calculation using the multilingual thesaurus eurovoc. In *Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing*, CICLing '02, pages 415–424, Berlin, Heidelberg, 2002. Springer-Verlag. ISBN 3540432191.

Sara Stymne, Jörg Tiedemann, and Joakim Nivre. Estimating word alignment quality for SMT reordering tasks. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 275–286, Baltimore, Maryland, USA, June 2014. Association for Computational Linguistics. doi: 10.3115/v1/W14-3334. URL `https://www.aclweb.org/anthology/W14-3334`.

Jinsong Su, Shan Wu, Deyi Xiong, Yaojie Lu, Xianpei Han, and Biao Zhang. Variational recurrent neural machine translation. *CoRR*, abs/1801.05119, 2018. URL `http://arxiv.org/abs/1801.05119`.

Ilya Sutskever, James Martens, and Geoffrey Hinton. Generating text with recurrent neural networks. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, pages 1017–1024, USA, 2011. Omnipress. ISBN 978-1-4503-0619-5. URL `http://dl.acm.org/citation.cfm?id=3104482.3104610`.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems (NIPS) 27*, pages 3104–3112. NIPS, 2014. URL `https://dl.acm.org/doi/10.5555/2969033.2969173`.

George Tambouratzis, Fotini Simistira, Sokratis Sofianopoulos, Nikos Tsimboukakis, and Marina Vassiliou. A resource-light phrase scheme for language-portable MT. In Mikel L. Forcada, Heidi Depraetere, and Vincent Vandeghinste, editors, *Proceedings of the 15th International Conference of the European Association for Machine Translation (EAMT)*, pages 185–192, 2011. URL `http://mt-archive.info/EAMT-2011-Tambouratzis.pdf`.

Akihiro Tamura, Taro Watanabe, and Eiichiro Sumita. Recurrent neural networks for word alignment model. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1470–1480, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

Tao Tao and ChengXiang Zhai. Mining comparable bilingual text corpora for cross-language information integration. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, KDD '05, pages 691–696, New York, NY, USA, 2005. Association for Computing Machinery. ISBN 159593135X. doi: 10.1145/1081870.1081958. URL `https://doi.org/10.1145/1081870.1081958`.

J. Tiedemann. *Bitext Alignment*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, 2011. ISBN 9781608455119. URL `https://dl.acm.org/citation.cfm?id=2031445`.

Jörg Tiedemann. *Recycling Translations – Extraction of Lexical Data from Parallel Corpora and their Application in Natural Language Processing*. PhD thesis, Uppsala University, Uppsala, Sweden, 2003. URL `http://uu.diva-portal.org/smash/record.jsf?pid=diva2:163715`. Anna Sågvall Hein, Åke Viberg (eds): Studia Linguistica Upsaliensia.

Jorg Tiedemann. Parallel data, tools and interfaces in opus. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Mehmet Ugur Dogan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may 2012. European Language Resources Association (ELRA). ISBN 978-2-9517408-7-7.

John Tinsley, Ventsislav Zhechev, Mary Hearne, and Andy Way. Robust language pair-independent sub-tree alignment. *Machine Translation Summit XI*, 2007. URL `http://www.mt-archive.info/MTS-2007-Tinsley.pdf`.

Nadi Tomeh. *Discriminative Alignment Models For Statistical Machine Translation*. Theses, Université Paris Sud - Paris XI, June 2012. URL `https://tel.archives-ouvertes.fr/tel-00720250`.

Kristina Toutanova and Michel Galley. Why initialization matters for IBM model 1: Multiple optima and non-strict convexity. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 461–466, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/P11-2081`.

Kristina Toutanova, H. Tolga Ilhan, and Christopher D. Manning. Extensions to hmm-based statistical word alignment models. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, EMNLP '02, pages 87–94, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1118693.1118705. URL `https://doi.org/10.3115/1118693.1118705`.

M. Ke Tran, Yonatan Bisk, Ashish Vaswani, Daniel Marcu, and Kevin Knight. Proceedings of the workshop on structured prediction for nlp. pages 63–71. Association for Computational Linguistics, 2016a.

M. Ke Tran, Yonatan Bisk, Ashish Vaswani, Daniel Marcu, and Kevin Knight. Unsupervised neural hidden markov models. In *Proceedings of the Workshop on Structured Prediction for NLP*, pages 63–71, Austin, TX, 2016b. Association for Computational Linguistics.

Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. Coverage-based neural machine translation. *CoRR*, abs/1601.04811, 2016a. URL `http://arxiv.org/abs/1601.04811`.

Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. Modeling coverage for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 76–85, Berlin, Germany, August 2016b. Association for Computational Linguistics. doi: 10.18653/v1/P16-1008. URL `https://www.aclweb.org/anthology/P16-1008`.

Takehito Utsuro, Hiroshi Ikeda, Masaya Yamane, Yuji Matsumoto, and Makoto Nagao. Bilingual text, matching using bilingual dictionary and statistics. In *COLING 1994 Volume 2: The 15th International Conference on Computational Linguistics*, 1994. URL `https://www.aclweb.org/anthology/C94-2175`.

Lonneke van der Plas and Jörg Tiedemann. Finding synonyms using automatic word alignment and measures of distributional similarity. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 866–873, Sydney, Australia, July 2006. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/P06-2111`.

D. Varga, Péter Halácsy, András Kornai, N. Viktor, N. László, and Tron Viktor. Parallel corpora for medium density languages. In *Recent Advances in Natural Language Processing IV: Selected papers from RANLP 2005*, 2007. URL `https://catalog.ldc.upenn.edu/docs/LDC2008T01/ranlp05.pdf`.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008, 2017. URL `http://papers.nips.cc/paper/7181-attention-is-all-you-need`.

Ashish Venugopal, Stephan Vogel, and Alex Waibel. Effective phrase translation extraction from alignment models. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 319–326, Sapporo, Japan, July 2003. Association for Computational Linguistics. doi: 10.3115/1075096.1075137. URL `https://www.aclweb.org/anthology/P03-1041`.

J. Véronis. *Parallel Text Processing: Alignment and Use of Translation Corpora*. Text, Speech and Language Technology. Springer, 2000. ISBN 9780792365464. URL `http://books.google.hr/books?id=I_4FPNS-RrEC`.

A. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. Inf. Theor.*, 13(2):260–269, September 1967. ISSN 0018-9448. doi: 10.1109/TIT.1967.1054010. URL https://doi.org/10.1109/TIT.1967.1054010.

Stephan Vogel, Hermann Ney, and Christoph Tillmann. Hmm-based word alignment in statistical translation. In *Proceedings of the 16th Conference on Computational Linguistics - Volume 2*, COLING '96, pages 836–841, Stroudsburg, PA, USA, 1996. Association for Computational Linguistics.

Thuy Vu, Ai Ti Aw, and Min Zhang. Feature-based method for document alignment in comparable news corpora. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 843–851, Athens, Greece, March 2009. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/E09-1096.

Changhan Wang, Kyunghyun Cho, and Jiatao Gu. Neural machine translation with byte-level subwords. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 9154–9160. AAAI Press, 2020. URL https://aaai.org/ojs/index.php/AAAI/article/view/6451.

Jianqiang Wang and Douglas W. Oard. *Matching Meaning for Cross-Language Information Retrieval*. PhD thesis, USA, 2005. URL https://dl.acm.org/doi/book/10.5555/1145159.

Peilu Wang, Yao Qian, Frank K. Soong, Lei He, and Hai Zhao. Part-of-speech tagging with bidirectional long short-term memory recurrent neural network. *CoRR*, abs/1510.06168, 2015a. URL http://arxiv.org/abs/1510.06168.

Weiyue Wang, Tamer Alkhouli, Derui Zhu, and Hermann Ney. Hybrid neural network alignment and lexicon model in direct HMM for statistical machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 125–131, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-2020. URL https://www.aclweb.org/anthology/P17-2020.

Weiyue Wang, Derui Zhu, Tamer Alkhouli, Zixuan Gan, and Hermann Ney. Neural hidden Markov model for machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 377–382, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-2060. URL https://www.aclweb.org/anthology/P18-2060.

Xiaolin Wang, Masao Utiyama, Andrew Finch, Taro Watanabe, and Eiichiro Sumita. Leave-one-out word alignment without garbage collector effects. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1817–1827, Lisbon, Portugal, September 2015b. Association for Computational Linguistics. doi: 10.18653/v1/D15-1209. URL https://www.aclweb.org/anthology/D15-1209.

Guillaume Wisniewski, Alexandre Allauzen, and François Yvon. Assessing phrase-based translation models with oracle decoding. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 933–943, Cambridge, MA, October 2010. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D/D10/D10-1091.

Guillaume Wisniewski, Anil Kumar Singh, and François Yvon. Quality estimation for machine translation: some lessons learned. *Machine Translation*, 27(3):213–238, 2013. ISSN 0922-6567. doi: 10.1007/s10590-013-9141-9. URL http://dx.doi.org/10.1007/s10590-013-9141-9.

Dekai Wu. Aligning a parallel English-Chinese corpus statistically with lexical criteria. In *32nd Annual Meeting of the Association for Computational Linguistics*, pages 80–87, Las Cruces, New Mexico, USA, June 1994. Association for Computational Linguistics. doi: 10.3115/981732.981744. URL https://www.aclweb.org/anthology/P94-1012.

Dekai Wu. Trainable coarse bilingual grammars for parallel text bracketing. In *Third Workshop on Very Large Corpora*, 1995. URL https://www.aclweb.org/anthology/W95-0106.

Dekai Wu. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3):377–403, 1997. URL https://www.aclweb.org/anthology/J97-3002.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144, 2016. URL http://arxiv.org/abs/1609.08144.

Jinxi Xu, Alexander Fraser, and Ralph Weischedel. Empirical studies in strategies for arabic retrieval. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '02, pages 269–274, New York, NY, USA, 2002. Association for Computing Machinery. ISBN 1581135610. doi: 10.1145/564376.564424. URL https://doi.org/10.1145/564376.564424.

Yong Xu. *Confidence Measures for Alignment and for Machine Translation*. PhD thesis, 2016. URL http://www.theses.fr/2016SACLS270/document. 2016SACLS270.

Kaoru Yamamoto, Taku Kudo, Yuta Tsuboi, and Yuji Matsumoto. Learning sequence-to-sequence correspondences from parallel corpora via sequential pattern mining. In *Proceedings of the HLT-NAACL 2003 Workshop on Building and Using Parallel Texts: Data Driven Machine Translation and Beyond*, pages 73–80, 2003. URL https://www.aclweb.org/anthology/W03-0314.

Nan Yang, Shujie Liu, Mu Li, Ming Zhou, and Nenghai Yu. Word alignment modeling with context dependent deep neural network. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 166–175. Association for Computational Linguistics, 2013.

Zichao Yang, Zhiting Hu, Yuntian Deng, Chris Dyer, and Alex Smola. Neural machine translation with recurrent attention modeling. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 383–387, Valencia, Spain, April 2017. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/E17-2061.

David Yarowsky, Grace Ngai, and Richard Wicentowski. Inducing multilingual text analysis tools via robust projection across aligned corpora. In *Proceedings of the First International Conference on Human Language Technology Research*, 2001. URL https://www.aclweb.org/anthology/H01-1035.

Wen-tau Yih, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao. Semantic parsing via staged query graph generation: Question answering with knowledge base. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1321–1331, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1128. URL https://www.aclweb.org/anthology/P15-1128.

T. Young, D. Hazarika, S. Poria, and E. Cambria. Recent Trends in Deep Learning Based Natural Language Processing. *ArXiv e-prints*, August 2017. URL https://arxiv.org/abs/1708.02709v8.

François Yvon, Yong Xu, Marianna Apidianaki, Clément Pillias, and Pierre Cubaud. Transread: Designing a bilingual reading experience with machine translation technologies. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 27–31, San Diego, California, June 2016.

Matthew D. Zeiler. ADADELTA: an adaptive learning rate method. *CoRR*, abs/1212.5701, 2012. URL http://arxiv.org/abs/1212.5701.

Thomas Zenkel, Joern Wuebker, and John DeNero. Adding interpretable attention to neural translation models improves word alignment. *CoRR*, abs/1901.11359, 2019. URL http://arxiv.org/abs/1901.11359.

Biao Zhang, Deyi Xiong, Jinsong Su, Hong Duan, and Min Zhang. Variational neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 521–530, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1050. URL https://www.aclweb.org/anthology/D16-1050.

Xiang Zhang and Yann LeCun. Text understanding from scratch. *CoRR*, abs/1502.01710, 2015. URL http://arxiv.org/abs/1502.01710.

Ying Zhang and Stephan Vogel. An efficient phrase-to-phrase alignment model for arbitrarily long phrase and large corpora. In *In Proceedings of the 10th Conference of the European Association for Machine Translation (EAMT-05*, pages 30–31. European Association for Machine Translation, 2005.

Yuqi Zhang, Evgeny Matusov, and Hermann Ney. Are unaligned words important for machine translation ? In *Conference of the European Association for Machine Translation*, pages 226–233, Barcelona, Spain, May 2009. URL http://www.mt-archive.info/EAMT-2009-Zhang.pdf.

Ventsislav Zhechev and Andy Way. Automatic generation of parallel treebanks. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 1105–1112, Manchester, UK, August 2008. Coling 2008 Organizing Committee. URL https://www.aclweb.org/anthology/C08-1139.

Chunting Zhou and Graham Neubig. Multi-space variational encoder-decoders for semi-supervised labeled sequence transduction. *CoRR*, abs/1704.01691, 2017. URL http://arxiv.org/abs/1704.01691.