



HAL
open science

Learning distributed representations of sentences using neural networks

Alexis Conneau

► **To cite this version:**

Alexis Conneau. Learning distributed representations of sentences using neural networks. Computer Science [cs]. Le Mans Université, 2019. English. NNT: . tel-02504543

HAL Id: tel-02504543

<https://hal.science/tel-02504543v1>

Submitted on 10 Mar 2020

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

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

THESE DE DOCTORAT DE

LE MANS UNIVERSITE
COMUE UNIVERSITE BRETAGNE LOIRE

ECOLE DOCTORALE N° 601
*Mathématiques et Sciences et Technologies
de l'Information et de la Communication*
Spécialité : Informatique

Par

Alexis Conneau

**« Apprentissage et applications de représentations multilingues
distribuées »**

Thèse présentée et soutenue à Paris, le 20 Mai 2019
Unité de recherche : Laboratoire d'Informatique de l'Université du Mans
Thèse N° : CIFRE 2016/0040 (Facebook)

Rapporteurs avant soutenance :

Claire Gardent Directrice de Recherche Première Classe au CNRS/LORIO (équipe SYNALP)
François Yvon Professeur des Universités à l'Université d'Orsay

Composition du Jury :

Attention, en cas d'absence d'un des membres du Jury le jour de la soutenance, la composition du jury doit être revue pour s'assurer qu'elle est conforme et devra être répercutée sur la couverture de thèse

Président :	Prénom Nom	Fonction et établissement d'exercice (9)(à préciser après la soutenance)
Examineurs :	Yann Lecun Chris Dyer	Professeur à New York University et Facebook AI Research Research Scientist at Google DeepMind
Dir. de thèse :	Paul Deléglise	Professeur émérite à l'Université du Mans
Co-dir. de thèse :	Loïc Barrault	Maître de conférence à l'Université du Mans
Co-dir. de thèse :	Holger Schwenk	Professeur à l'Université du Maine et Facebook AI Research

Titre : Apprentissage et applications de représentations distribuées multilingues

Mots clés : Apprentissage machine, réseaux de neurones profonds, représentations de phrases

Résumé : La capacité d'apprendre des représentations génériques d'objets tels que des images, des mots ou des phrases est essentielle pour construire des algorithmes qui ont une compréhension élargie du monde. Grâce à l'apprentissage par transfert, les réseaux neuronaux peuvent apprendre des représentations d'objets comme des images à partir de gros jeux de données, puis les exploiter pour améliorer la performance des tâches à faibles ressources. Bien que l'apprentissage par transfert ait été très efficace pour transférer les représentations d'images apprises sur ImageNet à des tâches de vision à faibles ressources, les représentations génériques de texte à l'aide de réseaux neuronaux se sont limitées aux représentations de mots. Cette thèse présente une étude des représentations de phrases. J'y présente comment l'on a poussé l'état de l'art des embeddings monolingues et cross-lingues. Les premières contributions de cette thèse incluent SentEval, un outil d'évaluation et d'analyse des représentations de phrases universelles et InferSent, un encodeur de phrases générique.

Nous montrons dans cette première partie que des représentations génériques de phrase peuvent être construites via des réseaux de neurones et qu'elles fournissent des caractéristiques (« features ») puissantes de phrases, utilisables dans de nombreux contextes. Dans la deuxième partie de ma thèse, mes contributions traitent de l'alignement de distributions de mots et de phrases dans plusieurs langues. Je montre pour la première fois qu'il est possible d'aligner des espaces de mots et de phrases de manière totalement non supervisée, sans aucune données parallèles. En particulier, nous montrons que nous pouvons traduire des mots de manière non supervisée, ce qui a été la pierre angulaire du nouveau domaine de recherche de "traduction automatique non supervisée". Ma dernière contribution sur la modélisation multilingue montre que les représentations de phrases provenant des modèles de langues peuvent être alignées de manière totalement non supervisée, ce qui conduit à un nouvel état de l'art en traduction automatique supervisée et non supervisée, et en classification cross-lingue.

Title : Learning distributed representations of sentences using neural networks

Keywords : Machine learning, deep neural networks, sentence embeddings

Abstract : Being able to learn generic representations of objects such as images, words or sentences is essential to building machines that have a broad understanding of the world. Through transfer learning, neural networks can learn representations from high-resource tasks and then leverage these to improve performance on low-resource task. While transfer learning has been very successful for transferring image features learned on ImageNet to low-resource visual understanding tasks, generic representations of text using neural networks were mostly limited to word embeddings. This dissertation presents a full study of sentence embeddings, through which I discuss how I have pushed the state of the art in monolingual and cross-lingual general-purpose embeddings. The first contributions of this thesis include SentEval, a transfer learning evaluation toolkit for universal sentence embeddings, InferSent, a state-of-the-art generic sentence encoder, and probing tasks, through which sentence encoders are analyzed and probed for linguistic properties.

We show in this first part that generic representations of sentence can be built and that they provide powerful out-of-the-box features of sentences. In the second part of my PhDs, my contributions have been centered around aligning distributions of words and sentences, in many languages. I show for the first time that it is possible to build generic cross-lingual word and sentence embedding spaces in a completely unsupervised way, without any parallel data. In particular, we show that we can perform word translation without parallel data, which was the building block for the new research field of "unsupervised machine translation". My last contribution on cross-lingual language modeling shows that state-of-the-art sentence representations can be aligned in a completely unsupervised way, leading to a new state of the art on supervised and unsupervised machine translation, and on the zero-shot crosslingual classification benchmarked called "XNLI".

Abstract

Being able to learn generic representations of objects such as images, words or sentences is essential to building machines that have a broad understanding of the world. Through transfer learning, neural networks can learn representations from high-resource tasks and then leverage these to improve performance on low-resource tasks. While transfer learning has been very successful for transferring image features learned on ImageNet to low-resource visual understanding tasks, generic representations of text using neural networks were mostly limited to word embeddings. This dissertation presents a full study of sentence embeddings, through which I discuss how I have pushed the state of the art in monolingual and cross-lingual general-purpose embeddings. The first contributions of this thesis include SentEval, a transfer learning evaluation toolkit for universal sentence embeddings, InferSent, a state-of-the-art generic sentence encoder, and probing tasks, through which sentence encoders are analyzed and probed for linguistic properties. We show in this first part that generic representations of sentences can be built and that they provide powerful out-of-the-box features of sentences. In the second part of my PhD, my contributions have been centered around aligning distributions of words and sentences, in many languages. I show for the first time that it is possible to build generic cross-lingual word and sentence embedding spaces in a completely unsupervised way, without any parallel data. In particular, we show that we can perform word translation without parallel data, which was the building block for the new research field of "unsupervised machine translation". The last contribution of this thesis on cross-lingual language modeling shows that state-of-the-art sentence representations can be aligned in a completely unsupervised way, leading to a new state of the art on supervised and unsupervised machine translation, and on the zero-shot cross-lingual natural language inference benchmarked dubbed "XNLI".

Acknowledgements

During my PhD, I have benefited from an exceptional research environment at Facebook AI Research in Paris where I have had the chance to work and interact with many great researchers and students. I owe this amazing opportunity to Yann Lecun who created FAIR, Florent Perronnin who managed the FAIR Paris lab and Holger Schwenk who gave me the opportunity. I am very grateful to them for having made FAIR Paris and my CIFRE PhD possible. I am also grateful to the company Facebook as a whole for creating such an incredible environment for young researchers.

I would like to express also my gratitude to Loic Barrault, my academic advisor at the University of Le Mans, for his enthusiasm and for giving me a lot of freedom in my research. The benevolence of Antoine Bordes and Hervé Jégou have been more than helpful during my PhD and I would like to particularly thank them for helping me grow as a researcher. I have also had the great chance to collaborate with Douwe Kiela, German Kruszewski, Marco Baroni, David Grangier, Michael Auli, Ludovic Denoyer and Marc'Aurelio Ranzato. I have learned a lot from working with these amazing researchers and I want to thank them for all they have taught me.

Finally, my interactions with other PhD students from Facebook AI Research in Paris have been essential and I want to thank especially the PhD crew. In particular, Timothée Lacroix has taught me a lot fustball-wise, and I am also thankful to Alexandre Sablayrolles, Pauline Luc, Neil Zeghidour, Alexandre Defossez, Rahma Chaabouni, Pierre Stock, Louis Martin, Angela Fan and others. They are all incredibly brilliant and interacting with them has been a blast. Special thanks goes to Patricia Le Carré for her continuous joie de vivre, for her support and for helping us so much along our PhDs at FAIR.

Last but not least, I want to thank my main collaborator Guillaume Lample. Our collaboration has been particularly fruitful and we have had a lot of fun when working together on our different projects. Working on MUSE was especially exciting and seeing our project presented at the internal F8 Facebook conference by our CTO Mike Schroepfer was incredibly rewarding. His enthusiasm for research and his creativity have been very inspiring.

Contents

Abstract	3
Acknowledgements	4
1 Introduction	1
2 Background	6
3 Monolingual sentence representations	14
3.1 Evaluating Sentence Embeddings	14
3.1.1 Introduction	15
3.1.2 Evaluations	17
3.1.3 Usage and Requirements	20
3.1.4 Baselines	23
3.1.5 Discussion	25
3.2 Learning General-Purpose Sentence Embeddings	26
3.2.1 Introduction	26
3.2.2 Related work	28
3.2.3 Approach	29
3.2.4 Empirical results	36
3.2.5 Visualization of BiLSTM-max sentence encoders	41
3.2.6 Conclusion	42
3.3 Probing Sentence Embeddings for Linguistic Properties	44
3.3.1 Introduction	44

3.3.2	Probing tasks	46
3.3.3	Sentence embedding models	49
3.3.4	Probing task experiments	52
3.3.5	Related work	58
3.3.6	Complementary analysis	60
3.3.7	Conclusion	63
4	Cross-lingual sentence representations	69
4.1	Aligning Word Representations Without Parallel Data	70
4.1.1	Introduction	70
4.1.2	Model	73
4.1.3	Training and architectural choices	77
4.1.4	Experiments	80
4.1.5	Complementary analysis	86
4.1.6	Related work	86
4.1.7	Conclusion	89
4.2	Evaluating Cross-Lingual Sentence Representations	91
4.2.1	Introduction	91
4.2.2	Related Work	93
4.2.3	The XNLI Corpus	95
4.2.4	Cross-Lingual NLI	98
4.2.5	Experiments and Results	102
4.2.6	Conclusion	105
4.3	Cross-lingual Language Models	106
4.3.1	Introduction	107
4.3.2	Related Work	108
4.3.3	Cross-lingual language models	108
4.3.4	Cross-lingual language model pretraining	111
4.3.5	Experiments and results	112
4.3.6	Conclusion	119
5	Conclusion	120

List of Tables

3.1	SentEval classification tasks	17
3.2	NLI and STS SentEval tasks	17
3.3	Transfer test results for various baseline methods	23
3.4	Evaluation of sentence representations on STS tasks	24
3.5	SentEval classification tasks	35
3.6	Impact of encoder architecture	35
3.7	Transfer test results for various architectures and objectives	36
3.8	COCO retrieval results	38
3.9	Source and target examples for seq2seq training tasks.	50
3.10	Probing task results	52
3.11	Test results for sentence encoder training tasks	61
3.12	Probing task accuracies with Logistic Regression.	65
3.13	Downstream tasks results for various sentence encoder architectures pre-trained in different ways.	66
4.1	Word translation retrieval P@1 of MUSE	80
4.2	English-Italian word translation	82
4.3	English-Italian sentence translation retrieval	84
4.4	Cross-lingual wordsim tasks	85
4.5	BLEU score on English-Esperanto	85
4.6	Unsupervised Esperanto-English word translation	86
4.7	XNLI examples (premise and hypothesis) from various languages	92
4.8	Average number of tokens per sentence in the XNLI corpus	96

4.9	BLEU scores of our translation models (XX-En) P@1 for multilingual word embeddings	102
4.10	XNLI test accuracy for the 15 languages	103
4.11	Ablation study of XNLI experiments	106
4.12	Results on cross-lingual classification accuracy	114
4.13	Results of XLM on unsupervised machine translation	116
4.14	Results of XLM on supervised machine translation	117
4.15	Results of XLM on low-resource language modeling	118
4.16	XLM unsupervised cross-lingual word embeddings	118

List of Figures

3.1	Generic NLI training scheme.	30
3.2	Bi-LSTM max-pooling network.	32
3.3	Inner Attention network architecture.	33
3.4	Hierarchical ConvNet architecture.	34
3.5	Transfer performance w.r.t. embedding size	37
3.6	Visualization of max-pooling for untrained BiLSTM-max (1)	42
3.7	Visualization of max-pooling for InferSent-4096 (1)	42
3.8	Visualization of max-pooling for untrained BiLSTM-max (2)	43
3.9	Visualization of max-pooling for InferSent-4096 (2)	43
3.10	Probing task scores after each training epoch, for NMT and SkipThought	57
3.11	Spearman correlation matrix between probing and downstream tasks	59
3.12	Evolution of probing tasks results wrt. embedding size	62
4.1	Toy illustration of MUSE	73
4.2	Unsupervised model selection	79
4.3	English to English word alignment accuracy	87
4.4	Illustration of language adaptation by sentence embedding alignment	101
4.5	Evolution of alignment losses and accuracies during training	104
4.6	Cross-lingual language model pretraining	110

Chapter 1

Introduction

Neural networks are very efficient tools that perform pattern matching and learn mappings from various inputs (images, text, speech) to output labels. Learning neural networks with millions of parameters on a particular task such as image classification requires a great amount of training data for the optimization algorithm to converge to a suitable minimum of the loss landscape. While this amount of data is available for some tasks - such as the ones where neural networks made a breakthrough like ImageNet (Deng et al., 2009) - it is simply impossible to label that many samples for each machine learning task. Instead, neural networks pretrained on large tasks are re-used on low-resource tasks either by learning a classifier on top of the output representations of the pretrained encoder or by fine-tuning both the encoder and the classifier on the new task (Oquab et al., 2014). This procedure is called transfer learning. While this has been rapidly successful in computer vision after the AlexNet breakthrough (Krizhevsky et al., 2012), transfer learning in natural language processing had been mostly limited to reusing pretrained word embeddings (Mikolov et al., 2013c). The goal of my thesis was to build general-purpose sentence encoders that provide embeddings that can be useful for a broad and diverse set of downstream tasks. I have made several contributions to the field of transfer learning in NLP during my PhD. The two main sections of this thesis correspond to the first and second half of my PhD in which I focused respectively on monolingual sentence representations and cross-lingual text representations. In the following, we make a brief description of those two chapters and of our contributions.

Monolingual Sentence Representations In the first part of this thesis, I describe how we can learn, evaluate and analyze universal sentence representations in the monolingual case. Precisely, I introduce SentEval (Conneau and Kiela, 2018), a framework that evaluates the generalization power of sentence embeddings through transfer learning on multiple downstream tasks. SentEval is the first toolkit that automates the evaluation of transfer learning for natural language processing. The user is only asked to provide a sentence encoder that takes text sentences as input and outputs sentence embeddings. For each of the downstream task, a logistic regression or a multi-Layer perceptron (MLP) is learned on top of the sentence embeddings on the training set of each task. Sentence embeddings are then defined as being "good" if they provide good features for a broad and diverse set of transfer tasks, that is if the dev/test accuracy on each downstream task is high. We open-sourced SentEval¹ to help the community build better sentence encoders. Together with SentEval, we made a thorough experimental exploration of various sentence encoders. Inspired by the success of transfer learning in computer vision where convolutional neural networks trained on ImageNet can be transferred to lower-resource tasks such as Pascal (Everingham et al., 2010), we explored various combinations of model architectures and training tasks. As opposed to computer vision, the natural language processing (NLP) community did not converge to a consensus regarding which model and which tasks should be combined to obtain the best general-purpose sentence representations. By evaluating multiple combinations of sentence encoder architectures and training tasks on SentEval, we built InferSent (Conneau et al., 2017), a new state-of-the-art sentence encoder that provide strong sentence representations. We show that it significantly outperforms previous approaches, including the average of pretrained word embeddings which is a powerful baseline, in particular for classification benchmarks. Our pretrained model provides sentence embeddings out of the box and was used extensively by the community after we open-sourced it. Finally, in this first chapter, we describe how we analyze the linguistic properties contained in the sentence embedding space. SentEval results help understand how powerful sentence embeddings are when used as features for tasks such as topic classification, sentiment analysis, natural language inference, semantic textual similarity or image-caption retrieval. But due to the complexity of those tasks, SentEval did not provide insights on what kinds of linguistic

¹t

properties are encoded in the sentence embedding space. Inspired by recurrent discussions in the community about "what can be crammed into fixed-size vectors", we built "probing tasks" (Conneau et al., 2018a). Each evaluating is designed for a single linguistic property of the input sentence, for example the subject number or genre. For instance, one probing task can be used to understand whether we can extract the sentence length using a logistic regression trained on top of the fixed-size sentence embeddings. Or probing tasks can be used to understand whether we can recover the information of words in the sentence, whether we can extract the number or gender of the main subject, or whether we can extract other grammatical information from the input sentence. Our ten probing tasks thus provide a convenient way to analyze and understand neural network sentence encoders, and we added them to our SentEval evaluation framework. We make a thorough analysis of the impact of the sentence encoder architecture and the impact of the training task on the linguistic properties contained in the sentence embedding space.

Cross-lingual sentence representations In the second part of my PhD, we extended our research from monolingual natural language understanding (NLU) to cross-lingual understanding (XLU). Indeed, the entire research community has been very English-centric and we wanted to depart from that by building representations for many languages. SentEval or the recently introduced GLUE benchmark (Wang et al., 2018) only include English benchmarks, and as InferSent requires human annotated data, it is not directly scalable to other languages. Even if InferSent only required unsupervised data in each language, it would not be convenient to have one model for each of the 7000 languages that exist in the world, or to carry 7000 machine translation systems that translate everything to English. As a result, in the second part of my PhD, I worked on aligning distributions and building cross-lingual text representations. As a first step, I built cross-lingual word embeddings (Conneau et al., 2018b). While the previous work of Mikolov et al. (2013b) had shown that a seed dictionary was needed to learn a linear mapping between two monolingual word2vec word embedding spaces in (say) English and French, we showed that we could completely remove the need for parallel supervision. Our work dubbed "MUSE" (for Multilingual Unsupervised and Supervised Embeddings) used in particular adversarial training to match the two distributions. Reducing the amount of supervision needed to align

embedding spaces is very important as cross-lingual understanding is particularly relevant for low-resource languages, for which by definition there exists little parallel data. In the shared word embedding space that we obtained after unsupervised alignment, looking at the nearest neighbor of a source word in the target space led to the correct word translation with approximately 70% accuracy for English and French. This means that using only monolingual corpora, we can exploit the similarity between languages to perform word translation in a completely unsupervised way. We are thus able to perform completely unsupervised word translation from one language to another, just by looking at monolingual data in the two languages. This line of work was further investigated by my colleagues and I, and this led to the field of "unsupervised machine translation" (Lample et al., 2018a,b), where we extended unsupervised machine translation from the word level to the sentence level. After the alignment of word embeddings, we extended this idea to the alignment of cross-lingual sentence representations. While SentEval and the GLUE benchmark focused exclusively on English natural language understanding tasks, we were interested in cross-lingual understanding tasks. In particular, as a way to evaluate cross-lingual sentence encoders, we built a zero-shot cross-lingual classification benchmark called the cross-lingual natural language inference (XNLI) benchmark (Conneau et al., 2018c). Our work was meant to catalyze research in the direction of cross-lingual understanding, which was partly sulked by the community due to the lack of evaluation benchmarks. The XNLI task is essential in practice as it represents a situation where training data is only available in a few high-resource languages, while predictions need to be made in many other languages, for instance to provide assistance for users. XNLI only provides English training data, but has dev and test sets in 14 other languages including two low-resources ones, Urdu and Swahili, for which cross-lingual transfer is particularly interesting. In this work, we aligned sentence encoders using an alignment loss that leverages millions of parallel data. Finally, the last work of my PhD focused on cross-lingual language models (Lample and Conneau, 2019). In this work we learn cross-lingual sentence representations that lead to a new state of the art on XNLI. Similar to the MUSE project on aligning distributions of words in an unsupervised way, we show that we can completely get rid of parallel data for aligning distributions of sentences while still obtaining very powerful cross-lingual sentence encoders. In this last article, we extend the approach of language model pretraining (Radford et al., 2018; Devlin et al.,

2018) to the cross-lingual setting to pretrain sentence encoders and decoders. We show that we can leverage parallel data to improve even more the alignment of our unsupervised approach. Using cross-lingual language model pretraining, we outperform by a large margin the previous state of the art in supervised and unsupervised neural machine translation, where we initialize both the encoders and the decoders, and in cross-lingual classification where we pretrain the sentence encoder and reach a new state of the art on XNLI. This work constitutes a significant step towards a better understanding of low-resource languages and multilinguality.

Chapter 2

Background

In this section, we summarize the principal developments in NLP related to this PhD thesis with a focus on neural-based approach. In particular, we describe word embeddings, sentence encoders, neural machine translation, language modeling, multi-task learning, cross-lingual understanding and transfer learning in NLP.

Transfer learning for natural language processing has been very successful only in recent years. In computer vision however, image representations obtained from convolutional neural networks (Fukushima, 1980; Waibel et al., 1995; LeCun et al., 1998) (CNNs) trained on ImageNet (Deng et al., 2009) had been successfully transferred to lower-resource tasks (Oquab et al., 2014). Indeed, learning CNNs amounts to estimating millions of parameters and requires a very large number of annotated image samples. This property prevents application of CNNs to problems with limited training data and makes transfer learning essential for most computer vision tasks.

For natural language processing, progress in learning and transferring representations of text was indeed not as striking. Universal text representations were mostly limited to word embeddings, the most successful approach being word2vec (Mikolov et al., 2013c,a), a fast method to learn word representations from a large unsupervised text corpus. Word2vec is inspired from the Cloze task (Taylor, 1953) and the distributional hypothesis from Harris (1954) which states that words that occur in the same contexts tend to have the same meaning. In word2vec, a simple linear layer is used together with a ranking loss to make embeddings of words that appear in similar contexts closer to each other in the embedding

space than randomly sampled pairs of words. Interestingly, a very similar approach had already been introduced in the work of Bengio et al. (2003) and Collobert and Weston (2008) where they used an additional hidden layer, but the important contribution of word2vec was to simplify their idea and make it accessible to a larger audience by open-sourcing a fast and efficient tool, and pretrained word embeddings. The word2vec approach, although limited to words, was a breakthrough in natural language processing and in particular in transfer learning for NLP. When trained on a sufficiently large amount of data like Wikipedia or CommonCrawl (Mikolov et al., 2017), word embeddings were not domain-specific anymore but universal enough that they could be used in many different contexts. And it only required raw text data. As an extension of word representations, the average of word embeddings (also known as bag of vectors) was shown to provide strong features for sentence and document classification (Le and Mikolov, 2014; Joulin et al., 2016). The word2vec approach was followed by a large body of work extending and exploiting this idea. In particular, it was explained in (Levy and Goldberg, 2014) that the word2vec algorithm named skipgram with negative sampling was implicitly factorizing a word-context matrix, whose cells are the pointwise mutual information (PMI) of the respective word and context pairs. More practically, (Bojanowski et al., 2017) extended word2vec with a method dubbed "fastText" that exploits character-level information of the words by using character n-grams embeddings which allowed to build word embeddings for unseen words. Mikolov et al. (2017) introduced additional training tricks to improve the quality of word representations and explored the limit of word2vec by learning word embeddings on the CommonCrawl dataset that contains tens of billions of words, showing a significant increase in performance on word analogy and word similarity. Together with the area of research focused on the learning part of word embeddings, a large body of work investigated how to evaluate the representations. Mikolov et al. (2013c) proposed to use of word similarity tasks to evaluate how the cosine similarity between word embeddings is correlated with human judgment. They also proposed the word analogy task (Mikolov et al., 2013a) as a stopping criterion for learning word embeddings. Evaluation benchmarks included relatedness (Budanitsky and Hirst, 2006; Agirre et al., 2009), similarity (Hill et al., 2016b), topical similarity (Hatzivassiloglou et al., 2001) and domain similarity (Turney, 2012). As it is the case for many research projects, evaluation is key to building better methods. Similar to

what has been done for building better word representations, in this thesis, we build strong evaluation benchmarks as well as new learning methods for general-purpose sentence representations.

Word2vec is an unsupervised method that requires unannotated text data. This is an essential property that makes it generalizable to almost any language. For example, Grave et al. (2018) build word embedding spaces for 157 languages by using CommonCrawl data. In the earlier work of (Mikolov et al., 2013b), it was shown that word embedding spaces in multiple languages could be aligned to form a single shared embedding space, where words that have similar meaning are close in the sense of the cosine distance. Specifically, they show that a linear mapping can be learned from a source space (say French) to a target space (say English) by minimizing the dot-product of the projected source word vectors with their corresponding target word vectors using a seed dictionary of word translations. In this shared embedding space, the nearest neighbor of a source word in the target space corresponds to its translation with high accuracy and the performance is usually evaluated using the precision@K. This was the first successful approach for learning cross-lingual distributed text representations. A series of articles improves this method, for example using an orthogonal constraint (Smith et al., 2017) or by reducing the amount of parallel data needed (Artetxe et al., 2017). This line of work that attempts to reduce the amount of supervision needed to align text distributions is essential. Indeed, one of the main motivations for cross-lingual understanding is to improve methods for low-resource languages by leveraging data in high-resource languages. But for these low-resource languages, by definition, the amount of parallel data with other languages is scarce. In our work (Conneau et al., 2018b), we show that we can completely remove the need for parallel supervision to align word embedding spaces by using an adversarial approach (Ganin et al., 2016). We show that we can translate words without parallel data. This idea is further extended in our work on unsupervised machine translation (Lample et al., 2018a) and for learning general-purpose cross-lingual sentence representations in a fully unsupervised way (Lample and Conneau, 2019).

While pretrained word embeddings were largely successful in many NLP applications, in particular for text classification tasks with limited training data, little had been done to extend word embeddings to sentence embeddings, beyond the average of word vectors.

This was the beginning of the use of neural networks to encode sentences in fixed-size vectors. After the seminal work of (Mikolov, 2012) on recurrent neural network (RNN) based language model, Sutskever et al. (2014); Cho et al. (2014b) revisited the power of RNNs and long short term memory (Hochreiter and Schmidhuber, 1997) as way to both encode and decode sentences. In the paper of Sutskever et al. (2014), a RNN encoder summarizes an input sentence into a fixed-size vector, and the conditional RNN language model decoder uses that context vector to generate the output translation word by word using greedy or beam search. This led to the creation of "neural machine translation", a new paradigm that allows to perform machine translation in an end-to-end manner with the so-called "encoder-decoder" or "sequence to sequence" neural approach, replacing the former phrase-based approach (Koehn et al., 2007). These models are learned through stochastic gradient descent (SGD) or Adam (Kingma and Ba, 2014) and backpropagation through time (BPTT). Sequence to sequence models have been largely improved since the work of Sutskever, with a number of strong contributions such as using subword units to counteract the large vocabulary softmax problem (Sennrich et al., 2015b; Wu et al., 2016), using an attention mechanism to improve performance for long sentences (Bahdanau et al., 2015), the use of sequence-level loss to alleviate the discrepancy between train and test time (Ranzato et al., 2015; Wu et al., 2016), exploiting monolingual data through back-translation (Sennrich et al., 2015a), and the use of convolutional neural networks (Gehring et al., 2017) or self-attention Transformer networks (Vaswani et al., 2017) instead of LSTMs. The re-discovery of the power of LSTMs and other contributions in neural machine translation have impacted the rest of the NLP research community, providing new methods for parsing (Vinyals et al., 2015a), image captioning (Vinyals et al., 2015b; Xu et al., 2015), sequence labeling (Lample et al., 2016) or language modeling (Jozefowicz et al., 2016). In parallel, time-delay neural networks (Waibel et al., 1995), also known as 1-dimensional convolutional neural networks, have been successful for encoding and decoding sentences, for example for document classification (Kalchbrenner et al., 2014; Kim, 2014; Zhang et al., 2015; Conneau et al., 2016) where architectures were inspired from the computer vision community (He et al., 2016), and for neural machine translation (Gehring et al., 2016, 2017). However, those neural architectures were only trained to perform one task at a time

and were meant to provide general-purpose representations of text. (Kiros et al., 2015) proposed a way to learn general-purpose sentence representations that could be used for many different tasks. They extended the idea of the skipgram approach to the sentence-level using a sequence to sequence model that take an input sentence and is trained to predict the previous or next sentence in a novel (Zhu et al., 2015). Once the encoder is trained, they use it as a feature generator and learn logistic regressions on top of the sentence representations to perform classification on several downstream tasks. They obtained better results than the representations resulting from the average of word embeddings (a.k.a bag of vectors). While bags of vectors are strong baselines, they are inherently limited as they do not capture word order or word context (a word has the same representations regardless of the sentence it belongs to) and SkipThought was a first attempt to incorporate more information in the embedding space. Our own InferSent (Conneau et al., 2017) approach extends SkipThought by looking for the best combination of model and task to obtain the best universal sentence encoder. But InferSent leverages supervised data, similar to what was done in computer vision where ConvNets were trained on ImageNet and eventually used as feature generator for lower-resource tasks. Our SentEval tool (Conneau and Kiela, 2018) also extends the work of (Kiros et al., 2015) by providing an automatic evaluation of sentence embeddings on more than 15 downstream tasks. The goal of SentEval was to provide a unified evaluation benchmark that would help catalyze research in the direction of learning sentence representations. Following SentEval, a similar approach was done by New York University (NYU) through the GLUE benchmark (Wang et al., 2018). Understanding what neural networks understand about natural language is an essential area of research in deep learning for NLP. While SentEval and GLUE involve downstream tasks such as sentence classification or semantic textual similarity from which it is difficult to infer what linguistic information the embeddings contain, SentEval also involves probing tasks (Conneau et al., 2018a), which are meant to understand the embeddings. Each probing task is designed to probe a single property of the embedding space, such as word order, word content or subject number. Some of them are inspired from the earlier work of Shi et al. (2016); Adi et al. (2017) who also used auxiliary prediction tasks to probe neural machine translation encoder and simple embedding methods. Conneau et al. (2018a) proposes a study of the impact of the encoder architecture and the training task on the quality of sentence embeddings. The

approaches mentioned above proposed ways to build general-purpose representations but did not investigate multi-task learning. Extending the ability of neural networks so that they perform more than one task at a time, Luong et al. (2015) proposed multi-task sequence to sequence learning where a single LSTM encoder is used together with multiple decoders trained with a skipthought, parsing and machine translation objective. They showed that they could have strong performance on all the tasks with only one encoder and even improve the BLEU score of a machine translation system using additional side objectives. With the idea that if an encoder is meant to be universal it should be trained to perform multiple tasks at once, Subramanian et al. (2018) exploits the approach of (Luong et al., 2015) to learn generic representations of sentences and extends our InferSent approach to multi-task learning. They show that they can obtain better representations by training on machine translation, parsing, natural language inference (Bowman et al., 2015) at once and by evaluating them on SentEval tasks. While the previously described trend was to leverage available supervised datasets to obtain better representations, the work of Radford et al. (2018); Devlin et al. (2018) revived language modeling as a way to learn strong representations of text in a completely unsupervised way. In particular the BERT approach of Devlin et al. (2018) uses masked language modeling (MLM) that significantly outperforms the classical causal language modeling (CLM) for pretraining Transformer networks. The main methodological difference between language model pretraining and the fixed-size representations proposed in (Kiros et al., 2015; Conneau et al., 2017; Subramanian et al., 2018) was that the language model encoders were fine-tuned entirely on downstream tasks, as opposed to frozen representations on top of which a classifier is learned (Conneau and Kiela, 2018). They showed strong gains in performance on the GLUE benchmark (Wang et al., 2018) by fine-tuning their pretrained LSTM/Transformer language models on sentiment analysis (Socher et al., 2013), natural language inference (Bowman et al., 2015; Williams et al., 2018), semantic textual similarity (Cer et al., 2017) and machine translation (Lample and Conneau, 2019). The work of Radford et al. (2018); Devlin et al. (2018) on language model fine-tuning was another breakthrough in transfer learning for natural language processing as they found a pretraining task that is completely unsupervised, a property that is particularly useful as it allows domain adaptation and large-scale learning

on huge amount of unsupervised data (Jozefowicz et al., 2016; Radford et al., 2019). Similar to multi-task learning, (Johnson et al., 2017b) proposed to extend the encoder decoder approach to the multilingual setting through multilingual training. Their approach builds a neural machine translation architecture with only one encoder and one decoder for many different languages. The encoder decoder architecture is trained simultaneously on multiple machine translation objectives corresponding to multiple language pairs. For instance the encoder can encode French sentences but also German sentences and decode them in English. They showed that they can encode sentences coming from multiple languages in a shared embedding space, and that having a multilingual encoder and decoder can help transfer representations from high-resource to low-resource language pairs and eventually improve BLEU score. They used a shared subword vocabulary to encode words coming from various languages and alphabets. This work was an important step towards better cross-lingual understanding. While Mikolov et al. (2013b) had shown the possibility of aligning word distributions, they showed that multiple languages could be encoded in a single shared embedding space with the same encoder. Together with this contribution in machine translation, a series of work attempted to align distributions of sentences, with the property that in this shared multilingual sentence embedding space, translations should have similar embeddings. Schwenk et al. (2017); Artetxe and Schwenk (2018); Eriguchi et al. (2018) leverage available parallel data and use the same multilingual neural machine translation approach than Johnson et al. (2017b) but without attention to learn a multilingual sentence encoder. Their encoder is then used to provide features for cross-lingual classification (Schwenk and Li, 2018; Conneau et al., 2018c), similar to what was done in (Conneau and Kiela, 2018) but in the cross-lingual setting. Conneau et al. (2018c) uses a ranking loss to align sentence encoders and introduces the cross-lingual natural language inference (XNLI) benchmark to alleviate the lack of resources in evaluating cross-lingual understanding and cross-lingual sentence representations in particular. For the XNLI task, training data is only available in English, but at test time the model is required to make predictions in 14 other languages such as French, German, Russian, Swahili or Urdu. This task is also known as zero-shot cross-lingual classification and has loads of applications in a production setting where training data is largely available in a few languages but not in other lower-resource languages where the task is still very important (for example for building

NLP educational systems for users that only understand the low-resource languages). Conneau et al. (2018c) investigates simpler and more efficient ways than machine translation for cross-lingual classification and provides the first large-scale cross-lingual classification benchmark. The work of Artetxe and Schwenk (2018); Conneau et al. (2018c) was largely outperformed on XNLI by our latest work on cross-lingual language models introduced in Lample and Conneau (2019), where we extended the approach of Radford et al. (2018); Devlin et al. (2018) to the cross-lingual setting. We showed strong improvements on XNLI but also on supervised and unsupervised machine translation (Conneau et al., 2018b; Lample et al., 2018a,b) when both encoders and decoders were pretrained with cross-lingual language models. Similar to what we had done in (Conneau et al., 2018b), we showed that it was possible to align sentence representations in a fully unsupervised way using masked language modeling, and we proposed an additional training objective that can still leverage parallel data dubbed translation language modeling (TLM). To build a cross-lingual encoder, we concatenate the Wikipedia corpora, learn a subword vocabulary and use the MLM objective as if we only had one language. To improve even more the alignment between various languages we use TLM which is a natural extension of the BERT objective (Devlin et al., 2018) where we concatenate two parallel sentences and learn a Transformer model to recover masked word in the input. When used in combination with MLM, TLM provides strong improvements on cross-lingual classification as it forces the Transformer to leverage the context in another language when predicting a masked word. This approach was the last piece of work that we did as part of my PhD. In this last contribution, we built a fully unsupervised state-of-the-art cross-lingual sentence encoder.

Chapter 3

Monolingual sentence representations

In this section, we present successively how we can evaluate, learn and analyze fixed-size English sentence embeddings. Specifically, we show how to evaluate sentence representations through transfer learning, by building SentEval¹ (Conneau and Kiela, 2018), an evaluation toolkit for universal sentence embeddings. Then, we conduct a thorough analysis of the impact of the sentence encoder architecture and the training task on the quality of the sentence representations and build InferSent² (Conneau et al., 2017), a universal sentence encoder that was state of the art at the time of publication and that had a significant impact in the research community. Finally, we analyze sentence embedding spaces coming from various encoders by building probing tasks which are each made to probe a single linguistic property of an input sentence, like sentence length, parse tree type, word order or subject number (Conneau et al., 2018a).

3.1 Evaluating Sentence Embeddings

In this first section, we introduce SentEval, a toolkit for evaluating the quality of universal sentence representations which forms the basis of the first part of my thesis. In this section, we present various sentence embeddings baselines and will introduce InferSent in section

¹<https://github.com/facebookresearch/SentEval>

²<https://github.com/facebookresearch/InferSent>

3.2. SentEval encompasses a variety of tasks, including binary and multi-class classification, natural language inference and sentence similarity. The set of tasks was selected based on what appears to be the community consensus regarding the appropriate evaluations for universal sentence representations. The toolkit comes with scripts to download and preprocess datasets, and an easy interface to evaluate sentence encoders. The aim is to provide a fairer, less cumbersome and more centralized way for evaluating sentence representations.

3.1.1 Introduction

Following the recent word embedding upheaval, one of NLP's next challenges has become the hunt for universal general-purpose sentence representations. What distinguishes these representations, or embeddings, is that they are not necessarily trained to perform well on one specific task. Rather, their value lies in their transferability, i.e., their ability to capture information that can be of use in any kind of system or pipeline, on a variety of tasks.

Word embeddings are particularly useful in cases where there is limited training data, leading to sparsity and poor vocabulary coverage, which in turn lead to poor generalization capabilities. Similarly, sentence embeddings (which are often built on top of word embeddings) can be used to further increase generalization capabilities, composing unseen combinations of words and encoding grammatical constructions that are not present in the task-specific training data. Hence, high-quality universal sentence representations are highly desirable for a variety of downstream NLP tasks.

The evaluation of general-purpose word and sentence embeddings has been problematic (Chiu et al., 2016; Faruqui et al., 2016), leading to much discussion about the best way to go about it³. On the one hand, people have measured performance on intrinsic evaluations, e.g. of human judgments of word or sentence similarity ratings (Agirre et al., 2012; Hill et al., 2016b) or of word associations (Vulić et al., 2017). On the other hand, it has been argued that the focus should be on downstream tasks where these representations would actually be applied (Ettinger et al., 2016; Nayak et al., 2016). In the case of sentence representations, there is a wide variety of evaluations available, many from before

³See also workshops on evaluating representations for NLP, e.g. RepEval: <https://repeval2017.github.io/>

the “embedding era”, that can be used to assess representational quality on that particular task. Over the years, something of a consensus has been established, mostly based on the evaluations in seminal papers such as SkipThought (Kiros et al., 2015), concerning what evaluations to use. Recent works in which various alternative sentence encoders are compared use a similar set of tasks (Hill et al., 2016a; Kiros et al., 2015).

Implementing pipelines for this large set of evaluations, each with its own peculiarities, is cumbersome and induces unnecessary wheel reinventions. Another well-known problem with the current status quo, where everyone uses their own evaluation pipeline, is that different preprocessing schemes, evaluation architectures and hyperparameters are used. The datasets are often small, meaning that minor differences in the evaluation setup may lead to very different outcomes, which implies that results reported in papers are not always fully comparable.

In order to overcome these issues, we introduce SentEval⁴: a freely available toolkit that makes it easy to evaluate universal sentence representation encoders on a large set of evaluation tasks that has been established by community consensus. The aim of SentEval is to make research on universal sentence representations fairer, less cumbersome and more centralized. To achieve this goal, SentEval encompasses the following:

- one central set of evaluations, based on what appears to be community consensus;
- one common evaluation pipeline with fixed standard hyperparameters, apart from those tuned on validation sets, in order to avoid discrepancies in reported results; and
- easy access for anyone, meaning: a straightforward interface in Python, and scripts necessary to download and preprocess the relevant datasets.

In addition, we provide examples of models, such as a simple bag-of-words model. These could potentially also be used to extrinsically evaluate the quality of word embeddings in NLP tasks.

⁴<https://github.com/facebookresearch/SentEval>

name	N	task	C	examples	label(s)
MR	11k	sentiment (movies)	2	“Too slow for a younger crowd , too shallow for an older one.”	neg
CR	4k	product reviews	2	“We tried it out christmas night and it worked great .”	pos
SUBJ	10k	subjectivity/objectivity	2	“A movie that doesn’t aim too high , but doesn’t need to.”	subj
MPQA	11k	opinion polarity	2	“don’t want”; “would like to tell”;	neg, pos
TREC	6k	question-type	6	“What are the twin cities ?”	LOC:city
SST-2	70k	sentiment (movies)	2	“Audrey Tautou has a knack for picking roles that magnify her [..]”	pos
SST-5	12k	sentiment (movies)	5	“nothing about this movie works.”	0

Table 3.1: **Classification tasks.** C is the number of classes and N is the number of samples.


name	N	task	output	premise	hypothesis	label
SNLI	560k	NLI	3	“A small girl wearing a pink jacket is riding on a carousel.”	“The carousel is moving.”	entailment
SICK-E	10k	NLI	3	“A man is sitting on a chair and rubbing his eyes”	“There is no man sitting on a chair and rubbing his eyes”	contradiction
SICK-R	10k	STS	[0, 5]	“A man is singing a song and playing the guitar”	“A man is opening a package that contains headphones”	1.6
STS14	4.5k	STS	[0, 5]	“Liquid ammonia leak kills 15 in Shanghai”	“Liquid ammonia leak kills at least 15 in Shanghai”	4.6
MRPC	5.7k	PD	2	“The procedure is generally performed in the second or third trimester.”	“The technique is used during the second and, occasionally, third trimester of pregnancy.”	paraphrase
COCO	565k	ICR	sim	 5	“A group of people on some horses riding through the beach.”	rank

Table 3.2: **Natural Language Inference and Semantic Similarity tasks.** NLI labels are contradiction, neutral and entailment. STS labels are scores between 0 and 5. PD=paraphrase detection, ICR=image-caption retrieval.

3.1.2 Evaluations

Our aim is to obtain general-purpose sentence embeddings that capture generic information, which should be useful for a broad set of tasks. To evaluate the quality of these representations, we use them as features in various transfer tasks. In the following, we describe these downstream tasks.

Binary and multi-class classification We use a set of binary classification tasks (see Table 3.5) that covers various types of sentence classification, including sentiment analysis (MR and both binary and fine-grained SST) (Pang and Lee, 2005; Socher et al., 2013), question-type (TREC) (Voorhees and Tice, 2000), product reviews (CR) (Hu and Liu, 2004), subjectivity/objectivity (SUBJ) (Pang and Lee, 2004) and opinion polarity (MPQA) (Wiebe et al., 2005). We generate sentence vectors and classifier on top, either in the form of a Logistic Regression or an MLP. For MR, CR, SUBJ and MPQA, we use nested 10-fold cross-validation, for TREC cross-validation and for SST standard validation.

Entailment and semantic relatedness We also include the SICK dataset (Marelli et al., 2014) for entailment (SICK-E), and semantic relatedness datasets including SICK-R and the STS Benchmark dataset (Cer et al., 2017). For semantic relatedness, which consists of predicting a semantic score between 0 and 5 from two input sentences, we follow the approach of Tai et al. (2015) and learn to predict the probability distribution of relatedness scores. SentEval reports Pearson and Spearman correlation. In addition, we include the SNLI dataset (Bowman et al., 2015), a collection of 570k human-generated English sentence pairs supporting the task of natural language inference (NLI), also known as recognizing textual entailment (RTE). NLI consists of predicting whether two input sentences are entailed, neutral or contradictory. SNLI was specifically designed to serve as a benchmark for evaluating text representation learning methods.

Semantic Textual Similarity While semantic relatedness requires training a model on top of the sentence embeddings, we also evaluate embeddings on the unsupervised SemEval tasks. These datasets include pairs of sentences taken from news articles, forum discussions, news conversations, headlines, image and video descriptions labeled with a similarity score between 0 and 5. The goal is to evaluate how the cosine distance between two sentences correlate with a human-labeled similarity score through Pearson and Spearman correlations. We include STS tasks from 2012 (Agirre et al., 2012), 2013⁶ (Agirre et al., 2013), 2014 (Agirre et al., 2014), 2015 (Agirre et al., 2015) and 2016 (Agirre et al.,

⁵Antonio Rivera - CC BY 2.0 - flickr

⁶Due to License issues, we do not include the SMT subtask.

2016). Each of these tasks includes several subtasks. SentEval reports both the average and the weighted average (by number of samples in each subtask) of the Pearson and Spearman correlations.

Paraphrase detection The Microsoft Research Paraphrase Corpus (MRPC) (Dolan et al., 2004) is composed of pairs of sentences which have been extracted from news sources on the Web. Sentence pairs have been human-annotated according to whether they capture a paraphrase/semantic equivalence relationship. We use the same approach as with SICK-E, except that our classifier has only 2 classes, i.e., the aim is to predict whether the sentences are paraphrases or not.

Caption-Image retrieval The caption-image retrieval task evaluates joint image and language feature models (Lai and Hockenmaier, 2014). The goal is either to rank a large collection of images by their relevance with respect to a given query caption (Image Retrieval), or ranking captions by their relevance for a given query image (Caption Retrieval). The COCO dataset provides a training set of 113k images with 5 captions each. The objective consists of learning a caption-image compatibility score $\mathcal{L}_{\text{cir}}(x, y)$ from a set of aligned image-caption pairs as training data. We use a pairwise ranking-loss $\mathcal{L}_{\text{cir}}(x, y)$:

$$\sum_y \sum_k \max(0, \alpha - s(Vy, Ux) + s(Vy, Ux_k)) + \sum_x \sum_{k'} \max(0, \alpha - s(Ux, Vy) + s(Ux, Vy_{k'})),$$

where (x, y) consists of an image y with one of its associated captions x , $(y_k)_k$ and $(y_{k'})_{k'}$ are negative examples of the ranking loss, α is the margin and s corresponds to the cosine similarity. U and V are learned linear transformations that project the caption x and the image y to the same embedding space. We measure Recall@K, with $K \in \{1, 5, 10\}$, i.e., the percentage of images/captions for which the corresponding caption/image is one of the first K retrieved; and median rank. We use the same splits as Karpathy and Fei-Fei (2015), i.e., we use 113k images (each containing 5 captions) for training, 5k images for validation and 5k images for test. For evaluation, following Karpathy and Fei-Fei (2015),

we split the 5k images in 5 random sets of 1k images on which we compute the mean $R@1$, $R@5$, $R@10$ and median (Med r) over the 5 splits. We include 2048-dimensional pretrained ResNet-101 (He et al., 2016) features for all images.

3.1.3 Usage and Requirements

Our evaluations comprise two different types of tasks: ones where we need to learn a model on top of the provided sentence representations (e.g. classification or regression) and ones where we simply take the cosine similarity between the two representations, as for the STS tasks. In the binary and multi-class classification tasks, we fit either a Logistic Regression classifier or an MLP with one hidden layer on top of the sentence representations. For the natural language inference tasks, where we are given two sentences u and v , we provide the classifier with the input $\langle u, v, |u - v|, u * v \rangle$. To fit the Pytorch models, we use Adam (Kingma and Ba, 2014), with a batch size 64. We tune the L2 penalty of the classifier with grid-search on the validation set. When using SentEval, two functions should be implemented by the user:

- `prepare(params, dataset)`: sees the whole dataset and applies any necessary preprocessing, such as constructing a lookup table of word embeddings (this function is optional); and
- `batcher(params, batch)`: given a batch of input sentences, returns an array of the sentence embeddings for the respective inputs.

The main `batcher` function allows the user to encode text sentences using any Python framework. For example, the `batcher` function might be a wrapper around a model written in Pytorch, TensorFlow, Theano, DyNet, or any other framework⁷. To illustrate the use, here is an example of what an evaluation script looks like, having defined the `prepare` and `batcher` functions:

```
import senteval
```

⁷Or any other programming language, as long as the vectors can be passed to, or loaded from, code written in Python.

```
se = senteval.engine.SE(
    params, batcher, prepare)
transfer_tasks = ['MR', 'CR']
results = se.eval(transfer_tasks)
```

Parameters Both functions make use of a `params` object, which contains the settings of the network and the evaluation. `SentEval` has several parameters that influence the evaluation procedure. These include the following:

- `task_path` (string, required): path to the data.
- `seed` (int): random seed for reproducibility.
- `batch_size` (int): size of minibatch of text sentences provided to batcher (sentences are sorted by length).
- `kfold` (int): k in the kfold-validation (default: 10).

The default config is:

```
params = {'task_path': PATH_TO_DATA,
          'usepytorch': True,
          'kfold': 10}
```

We also give the user the ability to customize the classifier used for the classification tasks.

Classifier To be comparable to the results published in the literature, users should use the following parameters for Logistic Regression:

```
params['classifier'] =
    {'nhid': 0, 'optim': 'adam',
     'batch_size': 64, 'tenacity': 5,
     'epoch_size': 4}
```

The parameters of the classifier include:

- `nhid` (int): number of hidden units of the MLP; if `nhid > 0`, a Multi-Layer Perceptron with one hidden layer and a Sigmoid nonlinearity is used.
- `optim` (str): classifier optimizer (default: adam).
- `batch_size` (int): batch size for training the classifier (default: 64).
- `tenacity` (int): stopping criterion; maximum number of times the validation error does not decrease.
- `epoch_size` (int): number of passes through the training set for one epoch.
- `dropout` (float): dropout rate in the case of MLP.

For use cases where there are multiple calls to `SentEval`, e.g when evaluating the sentence encoder at every epoch of training on the dev sets, we propose the following prototyping set of parameters, which will lead to slightly worse results but will make the evaluation significantly faster:

```
params['classifier'] =  
    {'nhid': 0, 'optim': 'rmsprop',  
     'batch_size': 128, 'tenacity': 3,  
     'epoch_size': 2}
```

You may also pass additional parameters to the `params` object which will further be accessible from the `prepare` and `batcher` functions (e.g a pretrained model).

Datasets In order to obtain the data and preprocess it so that it can be fed into `SentEval`, we provide the `get_transfer_data.bash` script in the data directory. The script fetches the different datasets from their known locations, unpacks them and preprocesses them. We tokenize each of the datasets with the MOSES tokenizer (Koehn et al., 2007) and convert all files to UTF-8 encoding. Once this script has been executed, the `task_path` parameter can be set to indicate the path of the data directory.

Requirements SentEval is written in Python. In order to run the evaluations, the user will need to install numpy, scipy and recent versions of pytorch and scikit-learn. In order to facilitate research where no GPUs are available, we offer for the evaluations to be run on CPU (using scikit-learn) where possible. For the bigger datasets, where more complicated models are often required, for instance STS Benchmark, SNLI, SICK-R and the image-caption retrieval tasks, we recommend to train pytorch models on a single GPU.

3.1.4 Baselines

Model	MR	CR	SUBJ	MPQA	SST-2	SST-5	TREC	MRPC	SICK-E
<i>Representation learning (transfer)</i>									
GloVe LogReg	77.4	78.7	91.2	87.7	80.3	44.7	83.0	72.7/81.0	78.5
GloVe MLP	77.7	79.9	92.2	88.7	82.3	45.4	85.2	73.0/80.9	79.0
fastText LogReg	78.2	80.2	91.8	88.0	82.3	45.1	83.4	74.4/82.4	78.9
fastText MLP	78.0	81.4	92.9	88.5	84.0	45.1	85.6	74.4/82.3	80.2
SkipThought	79.4	83.1	93.7	89.3	82.9	-	88.4	72.4/81.6	79.5
<i>Supervised methods directly trained for each task (no transfer)</i>									
SOTA	83.1 ¹	86.3 ¹	95.5 ¹	93.3 ¹	89.5 ²	52.4 ²	96.1 ²	80.4/85.9 ³	84.5 ⁴

Table 3.3: Transfer test results for various baseline methods. We include supervised results trained directly on each task (no transfer). Results ¹ correspond to AdaSent (Zhao et al., 2015), ² to BLSTM-2DCNN (Zhou et al., 2016b), ³ to TF-KLD (Ji and Eisenstein, 2013) and ⁴ to Illinois-LH system (Lai and Hockenmaier, 2014).

Several baseline models are evaluated in Table 3.3 and in Table 3.4. We include two of the baselines that were most successful at the time of publication of our approach:

- Continuous bag-of-words embeddings (average of word vectors). We consider the most commonly used pretrained word vectors available, namely the fastText (Mikolov et al., 2017) and the GloVe (Pennington et al., 2014) vectors trained on Common-Crawl.
- SkipThought vectors (Ba et al., 2016)

In addition to these methods, we include the results of current state-of-the-art methods for which both the encoder and the classifier are trained on each task (no transfer). For

GloVe and fastText bag-of-words representations, we report the results for Logistic Regression and Multi-Layer Perceptron (MLP). For the MLP classifier, we tune the dropout rate and the number of hidden units in addition to the L2 regularization. We do not observe any improvement over Logistic Regression for methods that already have a large embedding size (4800 for SkipThought). On most transfer tasks, supervised methods that are trained directly on each task still outperform transfer methods. Our hope is that SentEval will help the community build sentence representations with better generalization power that can outperform both the transfer and the supervised methods.

Model	SST'12	SST'13	SST'14	SST'15	SST'16	SICK-R	SST-B
<i>Representation learning (transfer)</i>							
GloVe BoW	52.1	49.6	54.6	56.1	51.4	79.9	64.7
fastText BoW	58.3	57.9	64.9	67.6	64.3	82.0	70.2
SkipThought-LN	30.8	24.8	31.4	31.0	-	85.8	72.1
Char-phrase	66.1	57.2	74.7	76.1	-	-	-
<i>Supervised methods directly trained for each task (no transfer)</i>							
PP-Proj	60.0 ¹	56.8 ¹	71.3 ¹	74.8 ¹	-	86.8 ²	-

Table 3.4: Evaluation of sentence representations on the semantic textual similarity benchmarks. Numbers reported are Pearson correlations x100. We use the average of Pearson correlations for STS'12 to STS'16 which are composed of several subtasks. Charagram-phrase numbers were taken from (Wieting et al., 2016). Results ¹ correspond to PP-Proj (Wieting et al., 2015) and ² from Tree-LSTM (Tai et al., 2015).

Comparison with recent developments

After SentEval was introduced, another benchmark for evaluating sentence encoders called "GLUE" was created (Wang et al., 2018). While SentEval focuses on the evaluation of fixed-size frozen sentence embeddings, the goal of the GLUE benchmark was to evaluate in particular the pretraining of sentence encoders. That is, instead of only learning a logistic regression on top of sentence embeddings, the entire encoder is fine-tuned on each downstream task. Note that one advantage of not fine-tuning sentence encoders is that the same encoder can be used for multiple tasks as opposed to having one fine-tuned encoder

per task. Also, GLUE focused mostly on high-resource downstream tasks that have hundreds of thousand of training samples while SentEval focuses on low-resource transfer with small downstream tasks such as MR, CR etc. An interesting direction of research that we describe in the 4th chapter of this thesis is language model fine-tuning which started with the ELMO (Peters et al., 2018) approach and was further investigated in Radford et al. (2018); Devlin et al. (2018). This line of work was evaluated on the GLUE benchmark. Our contribution in that research area is described in the last chapter of this thesis on cross-lingual language models.

3.1.5 Discussion

Universal sentence representations are a hot topic in NLP research. Making use of a generic sentence encoder allows models to generalize and transfer better, even when trained on relatively small datasets, which makes them highly desirable for downstream NLP tasks.

We introduced SentEval as a fair, straightforward and centralized toolkit for evaluating sentence representations. We have aimed to make evaluation as easy as possible: sentence encoders can be evaluated by implementing a simple Python interface, and we provide a script to download the necessary evaluation datasets. In section 3.2, we make a thorough evaluation of which model and which trained task should be used to obtain the best possible universal sentence encoder on SentEval, and introduce InferSent (Conneau et al., 2017). In section 3.3, we enrich SentEval with probing tasks (Conneau et al., 2018a) to provide tools to better understand how the encoder understands language. Our hope is that our toolkit will be used by the community in order to ensure that fully comparable results are published in research papers.

3.2 Learning General-Purpose Sentence Embeddings

In the previous chapter, we introduced SentEval, a toolkit to automatically evaluate the quality of sentence representations. Using this evaluation measure, in this chapter, we conduct a thorough experimental investigation of the best (model, task) combination to obtain state-of-the-art sentence embeddings, inspired by the successful (ConvNet, ImageNet) combination in computer vision. As seen in the Background section, many modern NLP systems rely on word embeddings, previously trained in an unsupervised manner on large corpora, as base features. Efforts to obtain embeddings for larger chunks of text, such as sentences, were however not so successful when we conducted the present study. Several attempts at learning unsupervised representations of sentences had not reached satisfactory enough performance to be widely adopted, and language model fine-tuning (Radford et al., 2018) was not yet discovered. In this chapter, we show how universal sentence representations trained using the supervised data of the Stanford Natural Language Inference datasets can consistently outperform previous unsupervised methods like SkipThought vectors (Kiros et al., 2015) on a wide range of transfer tasks. Much like how computer vision uses ImageNet to obtain features, which can then be transferred to other tasks, our work tends to indicate the suitability of natural language inference for transfer learning to other NLP tasks. As part of this work, we made our encoder publicly available.⁸

3.2.1 Introduction

Distributed representations of words (or word embeddings) (Bengio et al., 2003; Collobert et al., 2011; Mikolov et al., 2013c; Pennington et al., 2014) have shown to provide useful features for various tasks in natural language processing and computer vision. While there seems to be a consensus concerning the usefulness of word embeddings and how to learn them, this is not yet clear with regard to representations that carry the meaning of a full sentence. That is, how to capture the relationships among multiple words and phrases in a single vector remains an question to be solved.

In this chapter, we study the task of learning universal representations of sentences,

⁸<https://www.github.com/facebookresearch/InferSent>

i.e., a sentence encoder model that is trained on a large corpus and subsequently transferred to other tasks. Two questions need to be solved in order to build such an encoder, namely: what is the preferable neural network architecture; and how and on what task should such a network be trained. Following existing work on learning word embeddings, most current approaches consider learning sentence encoders in an unsupervised manner like SkipThought (Kiros et al., 2015) or FastSent (Hill et al., 2016a). Here, we investigate whether supervised learning can be leveraged instead, taking inspiration from previous results in computer vision, where many models are pretrained on the ImageNet (Deng et al., 2009) before being transferred. We compare sentence embeddings trained on various supervised tasks, and show that sentence embeddings generated from models trained on a natural language inference task reach the best results in terms of transfer accuracy. We hypothesize that the suitability of NLI as a training task is caused by the fact that it is a high-level understanding task that involves reasoning about the semantic relationships within sentences.

Unlike in computer vision, where convolutional neural networks are predominant, there are multiple ways to encode a sentence using neural networks. Hence, we investigate the impact of the sentence encoding architecture on representational transferability, and compare convolutional, recurrent and even simpler word composition schemes. Our experiments show that an encoder based on a bi-directional LSTM architecture with max pooling, trained on the Stanford Natural Language Inference (SNLI) dataset (Bowman et al., 2015), yielded state-of-the-art sentence embeddings (at the time of publication) compared to previous alternative unsupervised approaches like SkipThought or FastSent, while being much faster to train. We establish this finding on a broad and diverse set of transfer tasks that measures the ability of sentence representations to capture general and useful information.

3.2.2 Related work

In the following, we present a more detailed related work than in the background, focused on learning representations. Transfer learning using supervised features has been successful in several computer vision applications (Razavian et al., 2014). Striking examples include face recognition (Taigman et al., 2014) and visual question answering (Antol et al., 2015), where image features trained on ImageNet (Deng et al., 2009) and word embeddings trained on large unsupervised corpora are combined.

In contrast, at the time of publication, most approaches for sentence representation learning were unsupervised, which we argued was because the NLP community had not yet found the best supervised task for embedding the semantics of a whole sentence. Interestingly, while our work focused on going from unsupervised to supervised learning, later work (Radford et al., 2018; Devlin et al., 2018) showed amazing results using only language models on large-scale datasets. Another motivation for having an unsupervised method is that neural networks are very good at capturing the biases of the task on which they are trained, but can easily forget the overall information or semantics of the input data by specializing too much on these biases. Learning models on large unsupervised task makes it harder for the model to specialize. Littwin and Wolf (2016) showed that co-adaptation of encoders and classifiers, when trained end-to-end, can negatively impact the generalization power of image features generated by an encoder. They propose a loss that incorporates multiple orthogonal classifiers to counteract this effect.

Work on generating sentence embeddings range from models that compose word embeddings (Le and Mikolov, 2014; Arora et al., 2017; Wieting et al., 2015) to more complex neural network architectures. SkipThought vectors (Kiros et al., 2015) propose an objective function that adapts the skip-gram model for words (Mikolov et al., 2013c) to the sentence level. By encoding a sentence to predict the sentences around it, and using the features in a linear model, they were able to demonstrate good performance on 8 transfer tasks. They further obtained better results using layer-norm regularization of their model in (Ba et al., 2016). Hill et al. (2016a) showed that the task on which sentence embeddings are trained significantly impacts their quality. In this section, we consider SkipThought with layer normalization as our baseline. The results on SentEval were presented in the last section.

In addition to unsupervised methods, Hill et al. (2016a) included supervised training in their comparison—namely, on machine translation data (using the WMT’14 English/French and English/German pairs), dictionary definitions and image captioning data (see also Kiela et al. (2017)) from the COCO dataset (Lin et al., 2014). These models obtained significantly lower results compared to the unsupervised Skip-Thought approach.

Previous work had explored training sentence encoders on the SNLI corpus and applying them on the SICK corpus (Marelli et al., 2014), either using multi-task learning or pretraining (Mou et al., 2016; Bowman et al., 2015). The results were inconclusive and did not reach the same level as simpler approaches that directly learn a classifier on top of unsupervised sentence embeddings instead (Arora et al., 2017). To our knowledge, this work is the first attempt to fully exploit the SNLI corpus for building generic sentence encoders. As we show in our experiments, we are able to consistently outperform unsupervised approaches, even if our models are trained on much less (but human-annotated) data.

3.2.3 Approach

This work combines two research directions, which we describe in what follows. First, we explain how the NLI task can be used to train universal sentence encoding models using the SNLI task. We subsequently describe the architectures that we investigated for the sentence encoder, which, in our opinion, covers a suitable range of sentence encoders that were in use at the time of publication. Since then, Transformer networks have become dominant in the field. Specifically, we examined standard recurrent models such as LSTMs and GRUs, for which we investigate mean and max-pooling over the hidden representations; a self-attentive network that incorporates different views of the sentence (similar in idea to the Transformer self-attention mechanism); and a hierarchical convolutional network that can be seen as a tree-based method that blends different levels of abstraction.

The Natural Language Inference task

The SNLI dataset consists of 570k human-generated English sentence pairs, manually labeled with one of three categories: entailment, contradiction and neutral. It captures natural language inference, also known in previous incarnations as Recognizing Textual Entailment, and constitutes one of the largest high-quality labeled resources explicitly constructed in order to require understanding sentence semantics. We hypothesize that the semantic nature of NLI makes it a good candidate for learning universal sentence embeddings in a supervised way. That is, we aim to demonstrate that sentence encoders trained on natural language inference are able to learn sentence representations that capture universally useful features.

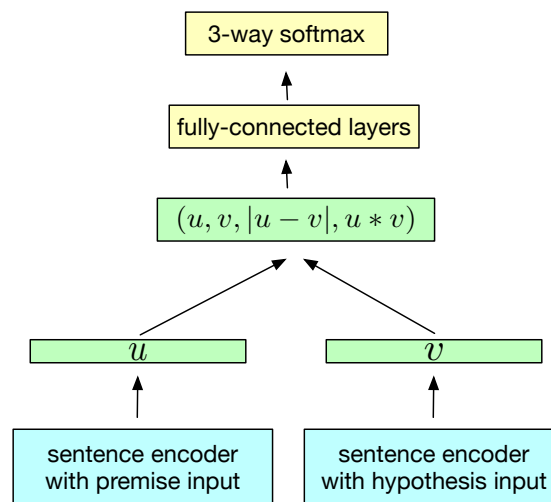


Figure 3.1: **Generic NLI training scheme.**

Models can be trained on SNLI in two different ways: (i) sentence encoding-based models that explicitly separate the encoding of the individual sentences and (ii) joint methods that allow to use encoding of both sentences (to use cross-features or attention from one sentence to the other).

Since our goal is to train a generic sentence encoder, we adopt the first setting. As illustrated in Figure 3.1, a typical architecture of this kind uses a shared sentence encoder that outputs a representation u for the premise and the v for the hypothesis. Once these sentence vectors are generated, three matching methods are applied to extract relations

between u and v : (i) concatenation of the two representations (u, v) ; (ii) element-wise product $u * v$; and (iii) absolute element-wise difference $|u - v|$. The resulting vector, which captures information from both the premise and the hypothesis, is fed into a 3-class classifier consisting of multiple fully-connected layers culminating in a softmax layer.

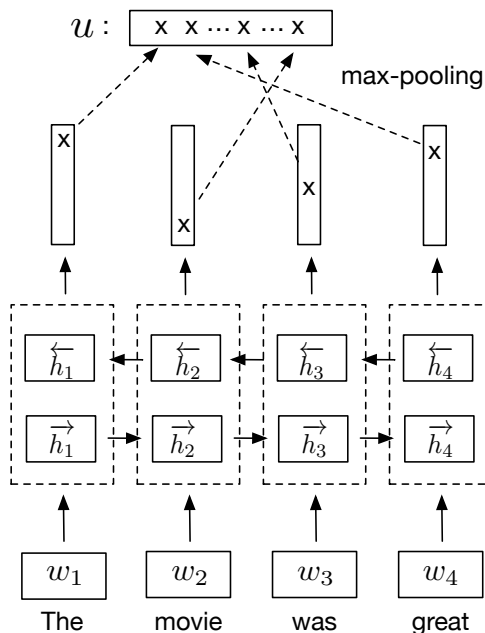
Sentence encoder architectures

A wide variety of neural networks for encoding sentences into fixed-size representations exists, and when it is not yet clear which one best captures generically useful information, even though there has been a recent consensus around Transformer networks (which did not exist at the time of this study). We compare 7 different architectures: standard recurrent encoders with either Long Short-Term Memory (LSTM) or Gated Recurrent Units (GRU), concatenation of last hidden states of forward and backward GRU, Bi-directional LSTMs (BiLSTM) with either mean or max pooling, self-attentive network and hierarchical convolutional networks.

LSTM and GRU Our first, and simplest, encoders apply recurrent neural networks using either LSTM (Hochreiter and Schmidhuber, 1997) or GRU (Cho et al., 2014a) modules, as in sequence to sequence encoders (Sutskever et al., 2014). For a sequence of T words (w_1, \dots, w_T) , the network computes a set of T hidden representations h_1, \dots, h_T , with $h_t = \overrightarrow{\text{LSTM}}(w_1, \dots, w_T)$ (or using GRU units instead). A sentence is represented by the last hidden vector, h_T .

We also consider a model BiGRU-last that concatenates the last hidden state of a forward GRU, and the last hidden state of a backward GRU to have the same architecture as for SkipThought vectors.

BiLSTM with mean/max pooling For a sequence of T words $\{w_t\}_{t=1, \dots, T}$, a bidirectional LSTM computes a set of T vectors $\{h_t\}_t$. For $t \in [1, \dots, T]$, h_t is the concatenation

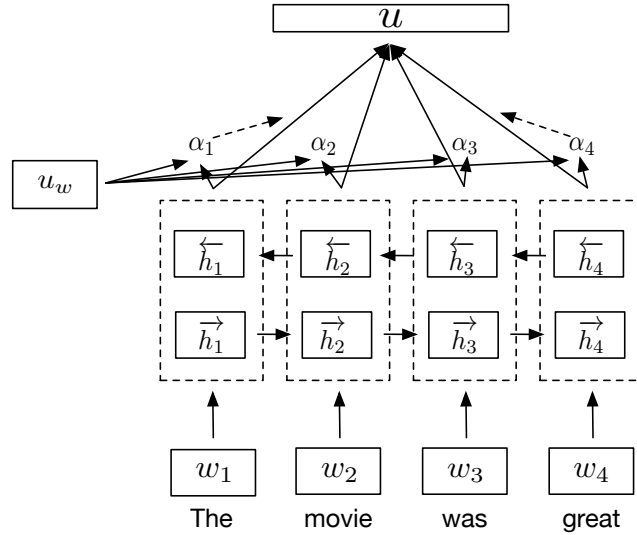
Figure 3.2: **Bi-LSTM max-pooling network.**

of a forward LSTM and a backward LSTM that read the sentences in two opposite directions:

$$\begin{aligned}\vec{h}_t &= \overrightarrow{\text{LSTM}}_t(w_1, \dots, w_T) \\ \overleftarrow{h}_t &= \overleftarrow{\text{LSTM}}_t(w_1, \dots, w_T) \\ h_t &= [\vec{h}_t, \overleftarrow{h}_t]\end{aligned}$$

We experiment with two ways of combining the varying number of $\{h_t\}_t$ to form a fixed-size vector, either by selecting the maximum value over each dimension of the hidden units (max pooling) (Collobert and Weston, 2008) or by considering the average of the representations (mean pooling).

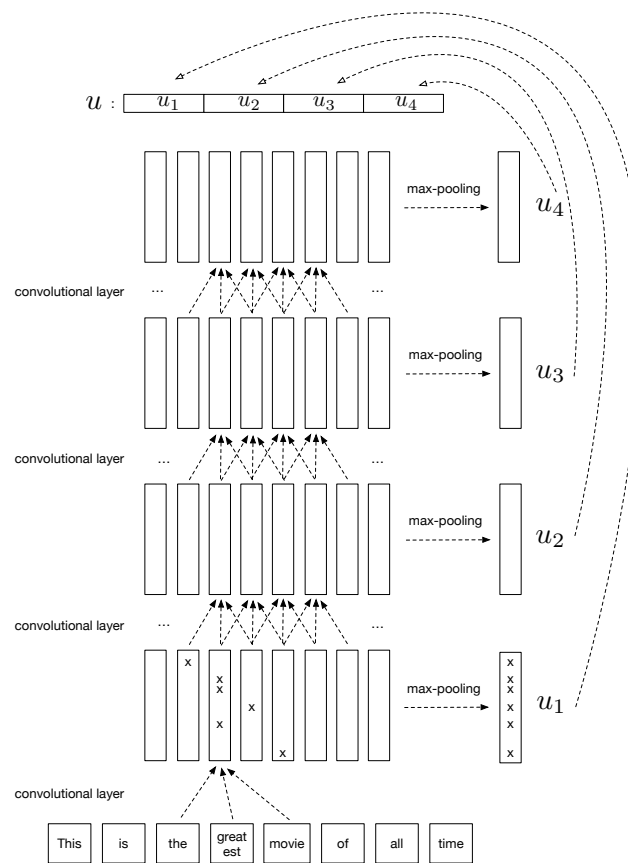
Self-attentive network The self-attentive sentence encoder (Liu et al., 2016; Lin et al., 2017) uses an attention mechanism over the hidden states of a BiLSTM to generate a representation u of an input sentence. The attention mechanism is defined as :

Figure 3.3: **Inner Attention network architecture.**

$$\begin{aligned}\bar{h}_i &= \tanh(W h_i + b_w) \\ \alpha_i &= \frac{e^{\bar{h}_i^T u_w}}{\sum_i e^{\bar{h}_i^T u_w}} \\ u &= \sum_t \alpha_i h_i\end{aligned}$$

where $\{h_1, \dots, h_T\}$ are the output hidden vectors of a BiLSTM. These are fed to an affine transformation (W, b_w) which outputs a set of keys $(\bar{h}_1, \dots, \bar{h}_T)$. The $\{\alpha_i\}$ represent the score of similarity between the keys and a learned context query vector u_w . These weights are used to produce the final representation u , which is a weighted linear combination of the hidden vectors.

Following Lin et al. (2017) we use a self-attentive network with multiple views of the input sentence, so that the model can learn which part of the sentence is important for the given task. Concretely, we have 4 context vectors $u_w^1, u_w^2, u_w^3, u_w^4$ which generate 4 representations that are then concatenated to obtain the sentence representation u . Figure 3.3 illustrates this architecture.

Figure 3.4: **Hierarchical ConvNet architecture.**

Hierarchical ConvNet One of the best performing models on classification tasks is a convolutional architecture termed *AdaSent* (Zhao et al., 2015), which concatenates different representations of the sentences at different level of abstractions. Inspired by this architecture, we introduce a faster version consisting of 4 convolutional layers. At every layer, a representation u_i is computed by a max-pooling operation over the feature maps (see Figure 3.4).

The final representation $u = [u_1, u_2, u_3, u_4]$ concatenates representations at different levels of the input sentence. The model thus captures hierarchical abstractions of an input sentence in a fixed-size representation.

Training details

For all our models trained on SNLI, we use SGD with a learning rate of 0.1 and a weight decay of 0.99. At each epoch, we divide the learning rate by 5 if the dev accuracy decreases. We use mini-batches of size 64 and training is stopped when the learning rate goes under the threshold of 10^{-5} . For the classifier, we use a multi-layer perceptron with 1 hidden-layer of 512 hidden units. We use open-source GloVe vectors trained on Common Crawl 840B with 300 dimensions as fixed word embeddings.

name	N	task	C	examples	label(s)
MR	11k	sentiment (movies)	2	“Too slow for a younger crowd , too shallow for an older one.”	neg
CR	4k	product reviews	2	“We tried it out christmas night and it worked great .”	pos
SUBJ	10k	subjectivity/objectivity	2	“A movie that doesn’t aim too high , but doesn’t need to.”	subj
MPQA	11k	opinion polarity	2	“don’t want”; “would like to tell”;	neg, pos
TREC	6k	question-type	6	“What are the twin cities ?”	LOC:city
SST-2	70k	sentiment (movies)	2	“Audrey Tautou has a knack for picking roles that magnify her [...]”	pos
SST-5	12k	sentiment (movies)	5	“nothing about this movie works.”	0

Table 3.5: **Classification tasks.** C is the number of classes and N is the number of samples.

Our aim is to obtain general-purpose sentence embeddings that capture generic information that is useful for a broad set of tasks. To evaluate the quality of these representations, we use them as features and evaluate them on 12 transfer tasks from SentEval.

Model	dim	NLI		Transfer	
		dev	test	micro	macro
LSTM	2048	81.9	80.7	79.5	78.6
GRU	4096	82.4	81.8	81.7	80.9
BiGRU-last	4096	81.3	80.9	82.9	81.7
BiLSTM-Mean	4096	79.0	78.2	83.1	81.7
Inner-attention	4096	82.3	82.5	82.1	81.0
HConvNet	4096	83.7	83.4	82.0	80.9
BiLSTM-Max	4096	85.0	84.5	85.2	83.7

Table 3.6: **Performance of sentence encoder architectures** on SNLI and (aggregated) transfer tasks. Dimensions of embeddings were selected according to best aggregated scores (see Figure 3.5). Underline means new state of the art on SNLI.

Model	MR	CR	SUBJ	MPQA	SST	TREC	MRPC	SICK-R	SICK-E	STS14
<i>Unsupervised representation training (unordered sentences)</i>										
Unigram-TFIDF	73.7	79.2	90.3	82.4	-	85.0	73.6/81.7	-	-	.58/.57
ParagraphVec (DBOW)	60.2	66.9	76.3	70.7	-	59.4	72.9/81.1	-	-	.42/.43
SDAE	74.6	78.0	90.8	86.9	-	78.4	73.7/80.7	-	-	.37/.38
SIF (GloVe + WR)	-	-	-	-	82.2	-	-	-	84.6	.69/ -
word2vec BOW [†]	77.7	79.8	90.9	88.3	79.7	83.6	72.5/81.4	0.803	78.7	.65/.64
fastText BOW [†]	78.3	81.0	92.4	87.8	81.9	84.8	73.9/82.0	0.815	78.3	.63/.62
GloVe BOW [†]	78.7	78.5	91.6	87.6	79.8	83.6	72.1/80.9	0.800	78.6	.54/.56
GloVe Positional Encoding [†]	78.3	77.4	91.1	87.1	80.6	83.3	72.5/81.2	0.799	77.9	.51/.54
BiLSTM-Max (untrained) [†]	77.5	81.3	89.6	88.7	80.7	85.8	73.2/81.6	0.860	83.4	.39/.48
<i>Unsupervised representation training (ordered sentences)</i>										
FastSent	70.8	78.4	88.7	80.6	-	76.8	72.2/80.3	-	-	.63/.64
FastSent+AE	71.8	76.7	88.8	81.5	-	80.4	71.2/79.1	-	-	.62/.62
SkipThought	76.5	80.1	93.6	87.1	82.0	92.2	73.0/82.0	0.858	82.3	.29/.35
SkipThought-LN	79.4	83.1	93.7	89.3	82.9	88.4	-	0.858	79.5	.44/.45
<i>Supervised representation training</i>										
CaptionRep (bow)	61.9	69.3	77.4	70.8	-	72.2	73.6/81.9	-	-	.46/.42
DictRep (bow)	76.7	78.7	90.7	87.2	-	81.0	68.4/76.8	-	-	.67/.70
NMT En-to-Fr	64.7	70.1	84.9	81.5	-	82.8	69.1/77.1	-	-	.43/.42
Paragram-phrase	-	-	-	-	79.7	-	-	0.849	83.1	.71/ -
BiLSTM-Max (on SST) [†]	(*)	83.7	90.2	89.5	(*)	86.0	72.7/80.9	0.863	83.1	.55/.54
BiLSTM-Max (on SNLI) [†]	79.9	84.6	92.1	89.8	83.3	88.7	75.1/82.3	0.885	86.3	.68/.65
BiLSTM-Max (on AllNLI) [†]	81.1	86.3	92.4	90.2	84.6	88.2	76.2/83.1	0.884	86.3	.70/.67
<i>Supervised methods (directly trained for each task – no transfer)</i>										
Naive Bayes - SVM	79.4	81.8	93.2	86.3	83.1	-	-	-	-	-
AdaSent	83.1	86.3	95.5	93.3	-	92.4	-	-	-	-
TF-KLD	-	-	-	-	-	-	80.4/85.9	-	-	-
Illinois-LH	-	-	-	-	-	-	-	-	84.5	-
Dependency Tree-LSTM	-	-	-	-	-	-	-	0.868	-	-

Table 3.7: **Transfer test results for various architectures trained in different ways.** Underlined are best results for transfer learning approaches, in bold are best results among the models trained in the same way. [†] indicates methods that we trained, other transfer models have been extracted from (Hill et al., 2016a). For best published supervised methods (no transfer), we consider AdaSent (Zhao et al., 2015), TF-KLD (Ji and Eisenstein, 2013), Tree-LSTM (Tai et al., 2015) and Illinois-LH system (Lai and Hockenmaier, 2014). (*) Our model trained on SST obtained 83.4 for MR and 86.0 for SST (MR and SST come from the same source), which we do not put in the tables for fair comparison with transfer methods.

3.2.4 Empirical results

In this section, we refer to ”micro” and ”macro” averages of development set (dev) results on transfer tasks whose metrics is accuracy: we compute a ”macro” aggregated score that corresponds to the classical average of dev accuracies, and the ”micro” score that is a sum

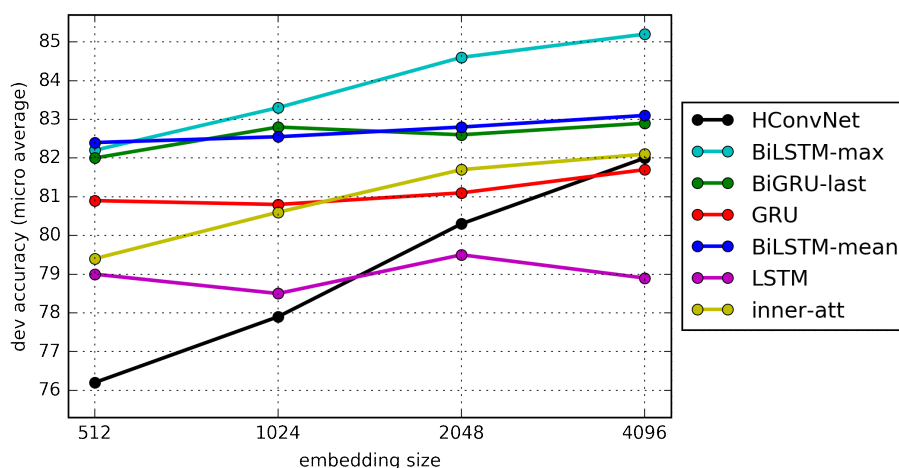


Figure 3.5: **Transfer performance w.r.t. embedding size** using the micro aggregation method.

of the dev accuracies, weighted by the number of dev samples.

Architecture impact

Model We observe in Table 3.6 that different models trained on the same NLI corpus lead to different transfer tasks results. The BiLSTM-4096 with the max-pooling operation performs best on both SNLI and transfer tasks. Looking at the micro and macro averages, we see that it performs significantly better than the other models LSTM, GRU, BiGRU-last, BiLSTM-Mean, inner-attention and the hierarchical-ConvNet.

Table 3.6 also shows that better performance on the training task does not necessarily translate in better results on the transfer tasks like when comparing inner-attention and BiLSTM-Mean for instance.

We hypothesize that some models are likely to over-specialize and adapt too well to the biases of a dataset without capturing general-purpose information of the input sentence. For example, the inner-attention model has the ability to focus only on certain parts of a sentence that are useful for the SNLI task, but not necessarily for the transfer tasks. On the other hand, BiLSTM-Mean does not make sharp choices on which part of the sentence is more important than others. The difference between the results seems to come from the different abilities of the models to incorporate general information while not focusing too

Model		Caption Retrieval				Image Retrieval			
		R@1	R@5	R@10	Med r	R@1	R@5	R@10	Med r
<i>Direct supervision of sentence representations</i>									
<i>m</i> -CNN	(Ma et al., 2015)	38.3	-	81.0	2	27.4	-	79.5	3
<i>m</i> -CNN _{ENS}	(Ma et al., 2015)	42.8	-	84.1	2	32.6	-	82.8	3
Order-embeddings	(Vendrov et al., 2016)	46.7	-	88.9	2	37.9	-	85.9	2
<i>Pre-trained sentence representations</i>									
SkipThought	+ VGG19 (82k)	33.8	67.7	82.1	3	25.9	60.0	74.6	4
SkipThought	+ ResNet101 (113k)	37.9	72.2	84.3	2	30.6	66.2	81.0	3
BiLSTM-Max (on SNLI)	+ ResNet101 (113k)	42.4	76.1	87.0	2	33.2	69.7	83.6	3
BiLSTM-Max (on AllNLI)	+ ResNet101 (113k)	42.6	75.3	87.3	2	33.9	69.7	83.8	3

Table 3.8: **COCO retrieval results.** SkipThought is trained either using 82k training samples with VGG19 features, or with 113k samples and ResNet-101 features (our setting). We report the average results on 5 splits of 1k test images.

much on specific features useful for the task at hand.

For a given model, the transfer quality is also sensitive to the optimization algorithm: when training with Adam instead of SGD, we observed that the BiLSTM-max converged faster on SNLI (5 epochs instead of 10), but obtained worse results on the transfer tasks, most likely because of the model and classifier’s increased capability to over-specialize on the training task.

Embedding size Figure 3.5 compares the overall performance of different architectures, showing the evolution of micro averaged performance with regard to the embedding size.

Since it is easier to linearly separate in high dimension, especially with logistic regression, it is not surprising that increased embedding sizes lead to increased performance for almost all models. However, this is particularly true for some models (BiLSTM-Max, HConvNet, inner-att), which demonstrate unequal abilities to incorporate more information as the size grows. We hypothesize that such networks are able to incorporate information that is not directly relevant to the objective task (results on SNLI are relatively stable with regard to embedding size) but that can nevertheless be useful as features for transfer tasks.

Task transfer

We report in Table 3.7 transfer tasks results for different architectures trained in different ways. We group models by the nature of the data on which they were trained. The first group corresponds to models trained with unsupervised unordered sentences. This includes bag-of-words models such as word2vec-SkipGram, the Unigram-TFIDF model, the Paragraph Vector model (Le and Mikolov, 2014), the Sequential Denoising Auto-Encoder (SDAE) (Hill et al., 2016a) and the SIF model (Arora et al., 2017), all trained on the Toronto book corpus (Zhu et al., 2015). The second group consists of models trained with unsupervised ordered sentences such as FastSent and SkipThought (also trained on the Toronto book corpus). We also include the FastSent variant “FastSent+AE” and the SkipThought-LN version that uses layer normalization. We report results from models trained on supervised data in the third group, and also report some results of supervised methods trained directly on each task for comparison with transfer learning approaches.

Comparison with SkipThought The best performing sentence encoder at the time of publication was the SkipThought-LN model, which was trained on a very large corpora of ordered sentences. With much less data (570k compared to 64M sentences) but with high-quality supervision from the SNLI dataset, we are able to consistently outperform the results obtained by SkipThought vectors. We train our model in less than a day on a single GPU compared to the best SkipThought-LN network trained for a month. Our BiLSTM-max trained on SNLI performs much better than the released SkipThought vectors on MR, CR, MPQA, SST, MRPC-accuracy, SICK-R, SICK-E and STS14 (see Table 3.7). Except for the SUBJ dataset, it also performs better than SkipThought-LN on MR, CR and MPQA. We also observe by looking at the STS14 results that the cosine metrics in our embedding space is much more semantically informative than in SkipThought embedding space (pearson score of 0.68 compared to 0.29 and 0.44 for ST and ST-LN). We hypothesize that this is namely linked to the matching method of SNLI models which incorporates a notion of distance (element-wise product and absolute difference) during training.

NLI as a supervised training set Our findings indicate that our model trained on SNLI obtains much better overall results than models trained on other supervised tasks such as

COCO, dictionary definitions, NMT, PPDB (Ganitkevitch et al., 2013) and SST. For SST, we tried exactly the same models as for SNLI; it is worth noting that SST is smaller than NLI. Our representations constitute higher-quality features for both classification and similarity tasks. One explanation is that the natural language inference task constrains the model to encode the semantic information of the input sentence, and that the information required to perform NLI is generally discriminative and informative.

Domain adaptation on SICK tasks Our transfer learning approach obtains better results than previous state-of-the-art on the SICK task - which can be seen as an out-domain version of SNLI - for both entailment and relatedness. We obtain a pearson score of 0.885 on SICK-R while (Tai et al., 2015) obtained 0.868, and we obtain 86.3% test accuracy on SICK-E while previous best hand-engineered models (Lai and Hockenmaier, 2014) obtained 84.5%. We also significantly outperformed previous transfer learning approaches on SICK-E (Bowman et al., 2015) that used the parameters of an LSTM model trained on SNLI to fine-tune on SICK (80.8% accuracy). We hypothesize that our embeddings already contain the information learned from the in-domain task, and that learning only the classifier limits the number of parameters learned on the small out-domain task.

Image-caption retrieval results In Table 3.8, we report results for the COCO image-caption retrieval task. We report the mean recalls of 5 random splits of 1K test images as in Ma et al. (2015); Vendrov et al. (2016). When trained with ResNet features and 30k more training data, the SkipThought vectors perform significantly better than the original setting, going from 33.8 to 37.9 for caption retrieval R@1, and from 25.9 to 30.6 on image retrieval R@1. Our approach pushes the results even further, from 37.9 to 42.4 on caption retrieval, and 30.6 to 33.2 on image retrieval. These results are comparable to previous approach of (Ma et al., 2015) that did not do transfer but directly learned the sentence encoding on the image-caption retrieval task. This supports the claim that pre-trained representations such as ResNet image features and our sentence embeddings can achieve competitive results compared to features learned directly on the objective task.

MultiGenre NLI The MultiNLI corpus (Williams et al., 2018) was recently released as a multi-genre version of SNLI. With 433K sentence pairs, MultiNLI improves upon SNLI in its coverage: it contains ten distinct genres of written and spoken English, covering most of the complexity of the language. We augment Table 3.7 with our model trained on both SNLI and MultiNLI (AllNLI). We observe a significant boost in performance overall compared to the model trained only on SNLI. Our model even reaches AdaSent performance on CR, suggesting that having a larger coverage for the training task helps learn even better general representations. On semantic textual similarity STS14, we are also competitive with PPDB based paragraph-phrase embeddings with a pearson score of 0.70. Interestingly, on caption-related transfer tasks such as the COCO image caption retrieval task, training our sentence encoder on other genres from MultiNLI does not degrade the performance compared to the model trained only SNLI (which contains mostly captions), which confirms the generalization power of our embeddings.

3.2.5 Visualization of BiLSTM-max sentence encoders

Our representations were trained to focus on parts of a sentence such that a classifier can easily tell the difference between contradictory, neutral or entailed sentences.

In Table 3.12 and Table 3.9, we investigate how the max-pooling operation selects the information from the hidden states of the BiLSTM, for our trained and untrained BiLSTM-max models (for both models, word embeddings are initialized with GloVe vectors).

For each time step t , we report the number of times the max-pooling operation selected the hidden state h_t (which can be seen as a sentence representation centered around word w_t).

Without any training, the max-pooling is rather even across hidden states, although it seems to focus consistently more on the first and last hidden states. When trained, the model learns to focus on specific words that carry most of the meaning of the sentence without any explicit attention mechanism.

Note that each hidden state also incorporates information from the sentence at different levels, explaining why the trained model also incorporates information from all hidden states.

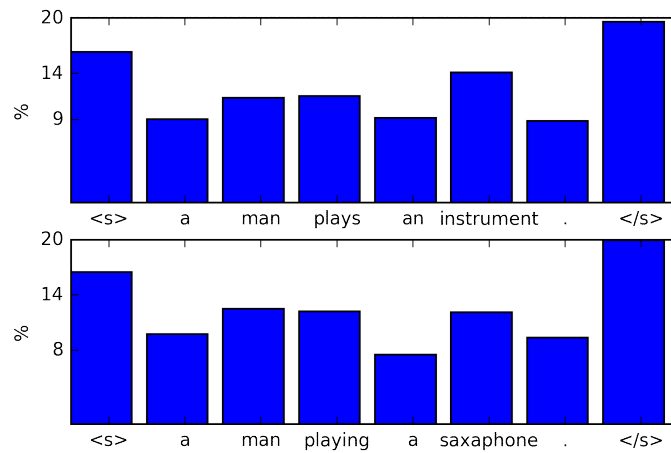


Figure 3.6: Pair of entailed sentences A: Visualization of max-pooling for BiLSTM-max 4096 **untrained**.

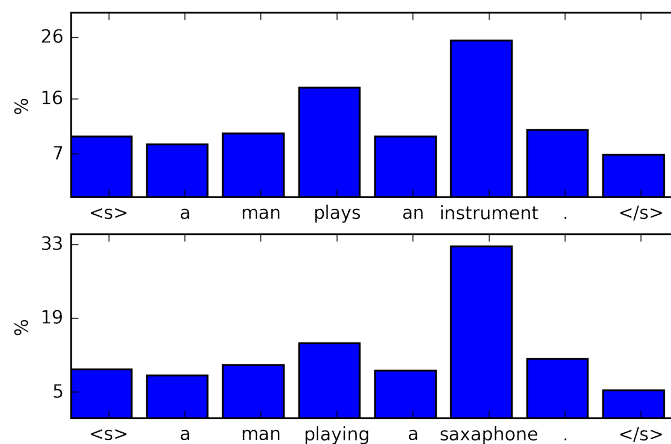


Figure 3.7: Pair of entailed sentences A: Visualization of max-pooling for BiLSTM-max 4096 **trained on NLI**.

3.2.6 Conclusion

This section studied the effects of training sentence embeddings with supervised data by testing on 12 different transfer tasks. We showed that models learned on NLI can perform better than models trained in unsupervised conditions such as SkipThought or on other supervised tasks. By exploring various architectures, we showed that a BiLSTM network with max pooling made the best universal sentence encoding method, outperforming previous

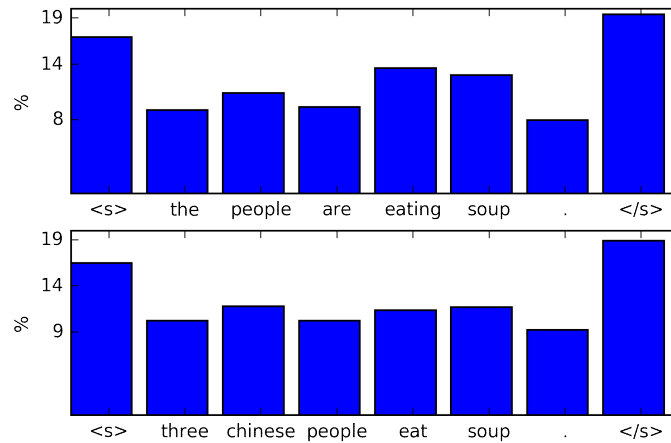


Figure 3.8: Pair of entailed sentences B: Visualization of max-pooling for BiLSTM-max 4096 **untrained**.

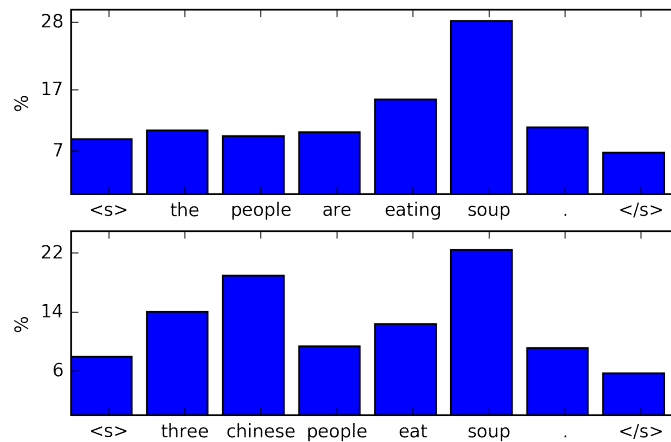


Figure 3.9: Pair of entailed sentences B: Visualization of max-pooling for BiLSTM-max 4096 **trained on NLI**.

approaches like SkipThought vectors.

We believe that this work only scratches the surface of possible combinations of models and tasks for learning generic sentence embeddings. Larger datasets that rely on natural language understanding for sentences could bring sentence embedding quality to the next level. Our work was in particular followed by multi-task learning (Subramanian et al., 2018) and the revival of language models (Radford et al., 2018; Devlin et al., 2018) for learning generic text representations in an unsupervised way.

3.3 Probing Sentence Embeddings for Linguistic Properties

In the previous sections, we have explored ways to evaluate and learn general-purpose sentence embeddings. In this last section about monolingual sentence representations, we complete our study by analyzing sentence embedding spaces. Although much effort has recently been devoted to training high-quality sentence embeddings, we still have a poor understanding of what they are capturing. “Downstream” tasks, often based on sentence classification, are commonly used to evaluate the quality of sentence representations as we have seen in the first chapter on SentEval. The complexity of the SentEval tasks makes it however difficult to infer what kind of information is present in the representations. In this section, we introduce 10 probing tasks designed to capture simple linguistic features of sentences, and we use them to study embeddings generated by three different encoders trained in eight distinct ways, uncovering intriguing properties of both encoders and training methods.

3.3.1 Introduction

Despite Ray Mooney’s quip that you cannot “cram the meaning of a whole sentence into a single vector”, sentence embedding methods have achieved impressive results in tasks ranging from machine translation (Sutskever et al., 2014; Cho et al., 2014b) to entailment detection (Williams et al., 2018), spurring the quest for “universal embeddings” trained once and used in a variety of applications (Kiros et al., 2015; Conneau et al., 2017; Subramanian et al., 2018). Positive results on concrete problems suggest that embeddings capture important linguistic properties of sentences. However, real-life “downstream” tasks require complex forms of inference, making it difficult to pinpoint the information a model is relying upon. Impressive as it might be that a system can tell that the sentence “A movie that doesn’t aim too high, but it doesn’t need to” (Pang and Lee, 2004) expresses a subjective viewpoint, it is hard to tell *how* the system (or even a human) comes to this conclusion. Complex tasks can also carry hidden biases that models might lock onto (Jabri et al., 2016). For example, Lai and Hockenmaier (2014) show that the simple heuristic of checking for

explicit negation words leads to good accuracy in the SICK sentence entailment task.

Model introspection techniques have been applied to sentence encoders in order to gain a better understanding of which properties of the input sentences their embeddings retain (see Section 3.3.5). However, these techniques often depend on the specifics of an encoder architecture, and consequently cannot be used to compare different methods. Shi et al. (2016) and Adi et al. (2017) introduced a more general approach, relying on the notion of what we will call *probing tasks*. A probing task is a classification problem that focuses on simple linguistic properties of sentences. For example, one such task might require to categorize sentences by the tense of their main verb. Given an encoder (e.g., an LSTM) pre-trained on a certain task (e.g., machine translation), we use the sentence embeddings it produces to train the tense classifier (without further embedding tuning). If the classifier succeeds, it means that the pre-trained encoder is storing readable tense information into the embeddings it creates. Note that:

1. The probing task asks a simple question, minimizing interpretability problems.
2. Because of their simplicity, it is easier to control for biases in probing tasks than in downstream tasks.
3. The probing task methodology is agnostic with respect to the encoder architecture, as long as it produces a vector representation of sentences.

We greatly extend earlier work on probing tasks as follows. First, we introduce a larger set of probing tasks (ten in total), organized by the type of linguistic properties they probe. Second, we systematize the probing task methodology, controlling for a number of possible nuisance factors, and framing all tasks so that they only require single sentence representations as input, for maximum generality and to ease result interpretation. Third, we use our probing tasks to explore a wide range of state-of-the-art encoding architectures and training methods, and further relate probing and downstream task performance. Finally, release our probing data sets and tools as part of SentEval, hoping they will become a standard way to study the linguistic properties of sentence embeddings.⁹

⁹<https://github.com/facebookresearch/SentEval/tree/master/data/probing>

3.3.2 Probing tasks

In constructing our probing benchmarks, we adopted the following criteria. First, for generality and interpretability, the task classification problem should only require single sentence embeddings as input (as opposed to, e.g., sentence and word embeddings, or multiple sentence representations). Second, it should be possible to construct large training sets in order to train parameter-rich multi-layer classifiers, in case the relevant properties are non-linearly encoded in the sentence vectors. Third, nuisance variables such as lexical cues or sentence length should be controlled for. Finally, and most importantly, we want tasks that address an interesting set of linguistic properties. We thus strove to come up with a set of tasks that, while respecting the previous constraints, probe a wide range of phenomena, from superficial properties of sentences such as which words they contain to their hierarchical structure to subtle facets of semantic acceptability. We think the current task set is reasonably representative of different linguistic domains, but we are not claiming that it is exhaustive. We expect future work to extend it.

The sentences for all our tasks are extracted from the Toronto Book Corpus (Zhu et al., 2015), more specifically from the random pre-processed portion made available by Paperno et al. (2016). We only sample sentences in the 5-to-28 word range. We parse them with the Stanford Parser (2017-06-09 version), using the pre-trained PCFG model (Klein and Manning, 2003), and we rely on the part-of-speech, constituency and dependency parsing information provided by this tool where needed. For each task, we construct training sets containing 100k sentences, and 10k-sentence validation and test sets. All sets are balanced, having an equal number of instances of each target class.

Surface information These tasks test the extent to which sentence embeddings are preserving surface properties of the sentences they encode. One can solve the surface tasks by simply looking at tokens in the input sentences: no linguistic knowledge is called for. The first task is to predict the *length* of sentences in terms of number of words (**SentLen**). Following Adi et al. (2017), we group sentences into 6 equal-width bins by length, and treat SentLen as a 6-way classification task. The *word content* (**WC**) task tests whether it is possible to recover information about the original words in the sentence from its embedding. We picked 1000 mid-frequency words from the source corpus vocabulary (the words

with ranks between 2k and 3k when sorted by frequency), and sampled equal numbers of sentences that contain one and only one of these words. The task is to tell which of the 1k words a sentence contains (1k-way classification). This setup allows us to probe a sentence embedding for word content without requiring an auxiliary word embedding (as in the setup of Adi and colleagues).

Syntactic information The next batch of tasks test whether sentence embeddings are sensitive to syntactic properties of the sentences they encode. The *bigram shift* (**BShift**) task tests whether an encoder is sensitive to legal word orders. In this binary classification problem, models must distinguish intact sentences sampled from the corpus from sentences where we inverted two random adjacent words (“What *you are* doing out there?”).

The *tree depth* (**TreeDepth**) task checks whether an encoder infers the hierarchical structure of sentences, and in particular whether it can group sentences by the depth of the longest path from root to any leaf. Since tree depth is naturally correlated with sentence length, we de-correlate these variables through a structured sampling procedure. In the resulting data set, tree depth values range from 5 to 12, and the task is to categorize sentences into the class corresponding to their depth (8 classes). As an example, the following is a long (22 tokens) but shallow (max depth: 5) sentence: “[₁ [₂ But right now, for the time being, my past, my fears, and my thoughts [₃ were [₄ my [₅business]]]..]” (the outermost brackets correspond to the ROOT and S nodes in the parse).

In the top constituent task (**TopConst**), sentences must be classified in terms of the sequence of top constituents immediately below the sentence (S) node. An encoder that successfully addresses this challenge is not only capturing latent syntactic structures, but clustering them by constituent types. TopConst was introduced by Shi et al. (2016). Following them, we frame it as a 20-way classification problem: 19 classes for the most frequent top constructions, and one for all other constructions. As an example, “[Then] [very dark gray letters on a black screen] [appeared] [.]” has top constituent sequence: “ADVP NP VP .”.

Note that, while we would not expect an untrained human subject to be explicitly aware of tree depth or top constituency, similar information must be implicitly computed to correctly parse sentences, and there is suggestive evidence that the brain tracks something akin

to tree depth during sentence processing (Nelson et al., 2017).

Semantic information These tasks also rely on syntactic structure, but they further require some understanding of what a sentence denotes. The **Tense** task asks for the tense of the main-clause verb (VBP/VBZ forms are labeled as present, VBD as past). No target form occurs across the train/dev/test split, so that classifiers cannot rely on specific words (it is not clear that Shi and colleagues, who introduced this task, controlled for this factor). The *subject number* (**SubjNum**) task focuses on the number of the subject of the main clause (number in English is more often explicitly marked on nouns than verbs). Again, there is no target overlap across partitions. Similarly, *object number* (**ObjNum**) tests for the number of the direct object of the main clause (again, avoiding lexical overlap). To solve the previous tasks correctly, an encoder must not only capture tense and number, but also extract structural information (about the main clause and its arguments). We grouped Tense, SubjNum and ObjNum with the semantic tasks, since, at least for models that treat words as unanalyzed input units (without access to morphology), they must rely on what a sentence denotes (e.g., whether the described event took place in the past), rather than on structural/syntactic information. We recognize, however, that the boundary between syntactic and semantic tasks is somewhat arbitrary.

In the *semantic odd man out* (**SOMO**) task, we modified sentences by replacing a random noun or verb o with another noun or verb r . To make the task more challenging, the bigrams formed by the replacement with the previous and following words in the sentence have frequencies that are comparable (on a log-scale) with those of the original bigrams. That is, if the original sentence contains bigrams $w_{n-1}o$ and ow_{n+1} , the corresponding bigrams $w_{n-1}r$ and rw_{n+1} in the modified sentence will have comparable corpus frequencies. No sentence is included in both original and modified format, and no replacement is repeated across train/dev/test sets. The task of the classifier is to tell whether a sentence has been modified or not. An example modified sentence is: “No one could see this Hayes and I wanted to know if it was real or a *spoonful* (orig.: *ploy*).” Note that judging plausibility of a syntactically well-formed sentence of this sort will often require grasping rather subtle semantic factors, ranging from selectional preference to topical coherence.

The coordination inversion (**CoordInv**) benchmark contains sentences made of two coordinate clauses. In half of the sentences, we inverted the order of the clauses. The task is to tell whether a sentence is intact or modified. Sentences are balanced in terms of clause length, and no sentence appears in both original and inverted versions. As an example, original “*They might be only memories, but I can still feel each one*” becomes: “I can still feel each one, but they might be only memories.” Often, addressing CoordInv requires an understanding of broad discourse and pragmatic factors.

Row **Hum. Eval.** of Table 3.10 reports human-validated “reasonable” upper bounds for all the tasks, estimated in different ways, depending on the tasks. For the surface ones, there is always a straightforward correct answer that a human annotator with enough time and patience could find. The upper bound is thus estimated at 100%. The TreeDepth, TopConst, Tense, SubjNum and ObjNum tasks depend on automated PoS and parsing annotation. In these cases, the upper bound is given by the proportion of sentences correctly annotated by the automated procedure. To estimate this quantity, one linguistically-trained we checked the annotation of 200 randomly sampled test sentences from each task. Finally, the BShift, SOMO and CoordInv manipulations can accidentally generate acceptable sentences. For example, one modified SOMO sentence is: “He pulled out the large round *onion* (orig.: *cork*) and saw the amber balm inside.”, that is arguably not more anomalous than the original. For these tasks, we ran Amazon Mechanical Turk experiments in which subjects were asked to judge whether 1k randomly sampled test sentences were acceptable or not. Reported human accuracies are based on majority voting.

3.3.3 Sentence embedding models

In this section, we present the three sentence encoders that we consider and the seven tasks on which we train them.

Sentence encoder architectures

A wide variety of neural networks encoding sentences into fixed-size representations exist. We focus here on three that have been shown to perform well on standard NLP tasks.

task	source	target
AutoEncoder	I myself was out on an island in the Swedish archipelago , at Sandhamn .	I myself was out on an island in the Swedish archipelago , at Sand@ ham@ n .
NMT En-Fr	I myself was out on an island in the Swedish archipelago , at Sandhamn .	Je me trouvais ce jour là sur une île de l' archipel suédois , à Sand@ ham@ n .
NMT En-De	We really need to up our particular contribution in that regard .	Wir müssen wirklich unsere spezielle Hilfs@ leistung in dieser Hinsicht aufstocken .
NMT En-Fi	It is too early to see one system as a universal panacea and dismiss another .	Nyt on liian aikaista nostaa yksi järjestelmä jal@ usta@ lle ja antaa jollekin toiselle huono arvo@ sana .
SkipThought	the old sami was gone , and he was a different person now .	the new sami didn 't mind standing barefoot in dirty white , sans ra@ y-@ bans and without beautiful women following his every move .
Seq2Tree	Dikoya is a village in Sri Lanka .	(ROOT (S (NP NNP)NP (VP VBZ (NP (NP DT NN)NP (PP IN (NP NNP NNP)NP)PP)NP)VP .)S)ROOT

Table 3.9: Source and target examples for seq2seq training tasks.

BiLSTM-last/max For a sequence of T words $\{w_t\}_{t=1,\dots,T}$, a bidirectional LSTM computes a set of T vectors $\{h_t\}_t$. For $t \in [1, \dots, T]$, h_t is the concatenation of a forward LSTM and a backward LSTM that read the sentences in two opposite directions. We experiment with two ways of combining the varying number of (h_1, \dots, h_T) to form a fixed-size vector, either by selecting the last hidden state of h_T or by selecting the maximum value over each dimension of the hidden units. The choice of these models are motivated by their demonstrated efficiency in seq2seq (Sutskever et al., 2014) and universal sentence representation learning (Conneau et al., 2017), respectively.¹⁰

Gated ConvNet We also consider the non-recurrent convolutional equivalent of LSTMs, based on stacked gated temporal convolutions. Gated convolutional networks were shown to perform well as neural machine translation encoders (Gehring et al., 2016) and language modeling decoders (Dauphin et al., 2016). The encoder is composed of an input word embedding table that is augmented with positional encodings (Sukhbaatar et al., 2015), followed by a stack of temporal convolutions with small kernel size. The output of each convolutional layer is filtered by a gating mechanism, similar to the one of LSTMs. Finally, max-pooling along the temporal dimension is performed on the output feature maps of the last convolution (Collobert and Weston, 2008).

¹⁰We also experimented with a unidirectional LSTM, with consistently poorer results.

Training tasks

Seq2seq systems have shown strong results in machine translation (Zhou et al., 2016a). They consist of an *encoder* that encodes a source sentence into a fixed-size representation, and a *decoder* which acts as a conditional language model and that generates the target sentence. We train **Neural Machine Translation** systems on three language pairs using about 2M sentences from the Europarl corpora (Koehn, 2005). We pick **English-French**, which involves two similar languages, **English-German**, involving larger syntactic differences, and **English-Finnish**, a distant pair. We also train with an **AutoEncoder** objective (Socher et al., 2011) on Europarl source English sentences. Following Vinyals et al. (2015a), we train a seq2seq architecture to generate linearized grammatical parse trees (see Table 3.9) from source sentences (**Seq2Tree**). We use the Stanford parser to generate trees for Europarl source English sentences. We train **SkipThought** vectors (Kiros et al., 2015) by predicting the next sentence given the current one (Tang et al., 2017), on 30M sentences from the Toronto Book Corpus, excluding those in the probing sets. Finally, as in InferSent in section 3.2, we train sentence encoders on **Natural Language Inference** using the concatenation of the SNLI (Bowman et al., 2015) and MultiNLI (Bowman et al., 2015) data sets (about 1M sentence pairs). In this task, a sentence encoder is trained to encode two sentences, which are fed to a classifier and whose role is to distinguish whether the sentences are contradictory, neutral or entailed. Finally, as in section 3.2, we also include **Untrained** encoders with random weights, which act as random projections of pre-trained word embeddings.

Training details

BiLSTM encoders use 2 layers of 512 hidden units (~ 4 M parameters), Gated ConvNet has 8 convolutional layers of 512 hidden units, kernel size 3 (~ 12 M parameters). We use pre-trained fastText word embeddings of size 300 (Mikolov et al., 2017) without fine-tuning, to isolate the impact of encoder architectures and to handle words outside the training sets. Training task performance and further details are in the subsection complementary analysis.

3.3.4 Probing task experiments

Task	SentLen	WC	TreeDepth	TopConst	BShift	Tense	SubjNum	ObjNum	SOMO	CoordInv
<i>Baseline representations</i>										
Majority vote	20.0	0.5	17.9	5.0	50.0	50.0	50.0	50.0	50.0	50.0
Hum. Eval.	100	100	84.0	84.0	98.0	85.0	88.0	86.5	81.2	85.0
Length	100	0.2	18.1	9.3	50.6	56.5	50.3	50.1	50.2	50.0
NB-uni-tfidf	22.7	97.8	24.1	41.9	49.5	77.7	68.9	64.0	38.0	50.5
NB-bi-tfidf	23.0	95.0	24.6	53.0	63.8	75.9	69.1	65.4	39.9	55.7
BoV-fastText	66.6	91.6	37.1	68.1	50.8	89.1	82.1	79.8	54.2	54.8
<i>BiLSTM-last encoder</i>										
Untrained	36.7	43.8	28.5	76.3	49.8	84.9	84.7	74.7	51.1	64.3
AutoEncoder	99.3	23.3	35.6	78.2	62.0	84.3	84.7	82.1	49.9	65.1
NMT En-Fr	83.5	55.6	42.4	81.6	62.3	88.1	89.7	89.5	52.0	71.2
NMT En-De	83.8	53.1	42.1	81.8	60.6	88.6	89.3	87.3	51.5	71.3
NMT En-Fi	82.4	52.6	40.8	81.3	58.8	88.4	86.8	85.3	52.1	71.0
Seq2Tree	94.0	14.0	59.6	89.4	78.6	89.9	94.4	94.7	49.6	67.8
SkipThought	68.1	35.9	33.5	75.4	60.1	89.1	80.5	77.1	55.6	67.7
NLI	75.9	47.3	32.7	70.5	54.5	79.7	79.3	71.3	53.3	66.5
<i>BiLSTM-max encoder</i>										
Untrained	73.3	88.8	46.2	71.8	70.6	89.2	85.8	81.9	73.3	68.3
AutoEncoder	99.1	17.5	45.5	74.9	71.9	86.4	87.0	83.5	73.4	71.7
NMT En-Fr	80.1	58.3	51.7	81.9	73.7	89.5	90.3	89.1	73.2	75.4
NMT En-De	79.9	56.0	52.3	82.2	72.1	90.5	90.9	89.5	73.4	76.2
NMT En-Fi	78.5	58.3	50.9	82.5	71.7	90.0	90.3	88.0	73.2	75.4
Seq2Tree	93.3	10.3	63.8	89.6	82.1	90.9	95.1	95.1	73.2	71.9
SkipThought	66.0	35.7	44.6	72.5	73.8	90.3	85.0	80.6	73.6	71.0
NLI	71.7	87.3	41.6	70.5	65.1	86.7	80.7	80.3	62.1	66.8
<i>GatedConvNet encoder</i>										
Untrained	90.3	17.1	30.3	47.5	62.0	78.2	72.2	70.9	61.4	59.6
AutoEncoder	99.4	16.8	46.3	75.2	71.9	87.7	88.5	86.5	73.5	72.4
NMT En-Fr	84.8	41.3	44.6	77.6	67.9	87.9	88.8	86.6	66.1	72.0
NMT En-De	89.6	49.0	50.5	81.7	72.3	90.4	91.4	89.7	72.8	75.1
NMT En-Fi	89.3	51.5	49.6	81.8	70.9	90.4	90.9	89.4	72.4	75.1
Seq2Tree	96.5	8.7	62.0	88.9	83.6	91.5	94.5	94.3	73.5	73.8
SkipThought	79.1	48.4	45.7	79.2	73.4	90.7	86.6	81.7	72.4	72.3
NLI	73.8	29.2	43.2	63.9	70.7	81.3	77.5	74.4	73.3	71.0

Table 3.10: **Probing task accuracies.** Classification performed by a MLP with sigmoid nonlinearity, taking pre-learned sentence embeddings as input (see complementary for details and logistic regression results).

Baselines Baseline and human-bound performance are reported in the top block of Table 3.10. **Length** is a linear classifier with sentence length as sole feature. **NB-uni-tfidf** is a Naive Bayes classifier using words’ tfidf scores as features, **NB-bi-tfidf** its extension to bigrams. Finally, **BoV-fastText** derives sentence representations by averaging the fastText

embeddings of the words they contain (same embeddings used as input to the encoders).¹¹

Except, trivially, for Length on SentLen and the NB baselines on WC, there is a healthy gap between top baseline performance and human upper bounds. NB-uni-tfidf evaluates to what extent our tasks can be addressed solely based on knowledge about the distribution of words in the training sentences. Words are of course to some extent informative for most tasks, leading to relatively high performance in Tense, SubjNum and ObjNum. Recall that the words containing the probed features are disjoint between train and test partitions, so we are not observing a confound here, but rather the effect of the redundancies one expects in natural language data. For example, for Tense, since sentences often contain more than one verb in the same tense, NB-uni-tfidf can exploit non-target verbs as cues: the NB features most associated to the past class are verbs in the past tense (e.g “sensed”, “lied”, “announced”), and similarly for present (e.g “uses”, “chuckles”, “frowns”). Using bigram features (NB-bi-tfidf) brings in general little or no improvement with respect to the unigram baseline, except, trivially, for the BShift task, where NB-bi-tfidf can easily detect unlikely bigrams. NB-bi-tfidf has below-random performance on SOMO, confirming that the semantic intruder is not given away by superficial bigram cues.

Our first striking result is the good overall performance of Bag-of-Vectors, confirming early insights that aggregated word embeddings capture surprising amounts of sentence information (Pham et al., 2015b; Arora et al., 2017; Adi et al., 2017). BoV’s good WC and SentLen performance was already established by Adi et al. (2017). Not surprisingly, word-order-unaware BoV performs randomly in BShift and in the more sophisticated semantic tasks SOMO and CoordInv. More interestingly, BoV is very good at the Tense, SubjNum, ObjNum, and TopConst tasks (much better than the word-based baselines), and well above chance in TreeDepth. The good performance on Tense, SubjNum and ObjNum has a straightforward explanation we have already hinted at above. Many sentences are naturally “redundant”, in the sense that most tensed verbs in a sentence are in the same tense, and similarly for number in nouns. In 95.2% Tense, 75.9% SubjNum and 78.7% ObjNum test sentences, the target tense/number feature is also the majority one for the whole sentence. Word embeddings capture features such as number and tense (Mikolov

¹¹Similar results are obtained summing embeddings, and using GloVe embeddings (Pennington et al., 2014).

et al., 2013d), so aggregated word embeddings will naturally track these features’ majority values in a sentence. BoV’s TopConst and TreeDepth performance is more surprising. Accuracy is well above NB, showing that BoV is exploiting cues beyond specific words strongly associated to the target classes. We conjecture that more abstract word features captured by the embeddings (such as the part of speech of a word) might signal different syntactic structures. For example, sentences in the “WHNP SQ .” top constituent class (e.g., “*How* long before you leave us again?”) must contain a wh word, and will often feature an auxiliary or modal verb. BoV can rely on this information to noisily predict the correct class.

Encoding architectures Comfortingly, proper encoding architectures clearly outperform BoV. An interesting observation in Table 3.10 is that different encoder architectures trained with the same objective, and achieving similar performance on the training task,¹² can lead to linguistically different embeddings, as indicated by the probing tasks. Coherently with the findings of Conneau et al. (2017) for the downstream tasks, this suggests that the prior imposed by the encoder architecture strongly preconditions the nature of the embeddings. Complementing recent evidence that convolutional architectures are on a par with recurrent ones in seq2seq tasks (Gehring et al., 2016), we find that Gated ConvNet’s overall probing task performance is comparable to that of the best LSTM architecture (although, as shown in the complementary analysis, the LSTM has a slight edge on downstream tasks). We also replicate the finding of section 3.2 that BiLSTM-max outperforms BiLSTM-last both in the downstream tasks (see below) and in the probing tasks (Table 3.10). Interestingly, the latter only outperforms the former in SentLen, a task that captures a superficial aspect of sentences (how many words they contain), that could get in the way of inducing more useful linguistic knowledge.

Training tasks We focus next on how different training tasks affect BiLSTM-max, but the patterns are generally representative across architectures. NMT training leads to encoders that are more linguistically aware than those trained on the NLI data set, despite the fact that we confirm the finding of section 3.2 that NLI is best for downstream tasks.

¹²See complementary analysis subsection for details on training task performance.

Perhaps, NMT captures richer linguistic features useful for the probing tasks, whereas shallower or more *ad-hoc* features might help more in our current downstream tasks. Suggestively, the one task where NLI clearly outperforms NMT is WC. Thus, NLI training is better at preserving shallower word features that might be more useful in downstream tasks (cf. Figure 3.11 and discussion there).

Unsupervised training (SkipThought and AutoEncoder) is not on a par with supervised tasks, but still effective. AutoEncoder training leads, unsurprisingly, to a model excelling at SentLen, but it attains low performance in the WC prediction task. This curious result might indicate that the latter information is stored in the embeddings in a complex way, not easily readable by our MLP. At the other end, Seq2Tree is trained to predict annotation from the same parser we used to create some of the probing tasks. Thus, its high performance on TopConst, Tense, SubjNum, ObjNum and TreeDepth is probably an artifact. Indeed, for most of these tasks, Seq2Tree performance is above the human bound, that is, Seq2Tree learned to mimic the parser errors in our benchmarks. For the more challenging SOMO and CoordInv tasks, that only indirectly rely on tagging/parsing information, Seq2Tree is comparable to NMT, that does not use explicit syntactic information.

Perhaps most interestingly, BiLSTM-max already achieves very good performance without any training (Untrained row in Table 3.10). Untrained BiLSTM-max also performs quite well in the downstream tasks. This architecture must encode priors that are intrinsically good for sentence representations. Untrained BiLSTM-max exploits the input fastText embeddings, and multiplying the latter by a random recurrent matrix provides a form of positional encoding. However, good performance in a task such as SOMO, where BoV fails and positional information alone should not help (the intruder is randomly distributed across the sentence), suggests that other architectural biases are at work. Intriguingly, a preliminary comparison of untrained BiLSTM-max and human subjects on the SOMO sentences evaluated by both reveals that, whereas humans have a bias towards finding sentences acceptable (62% sentences are rated as untampered with, vs. 48% ground-truth proportion), the model has a strong bias in the opposite direction (it rates 83% of the sentences as modified). A cursory look at contrasting errors confirms, unsurprisingly, that those made by humans are perfectly justified, while model errors are opaque. For example, the sentence “I didn’t come here to *reunite* (orig. *undermine*) you” seems perfectly

acceptable in its modified form, and indeed subjects judged it as such, whereas untrained BiLSTM-max “correctly” rated it as a modified item. Conversely, it is difficult to see any clear reason for the latter tendency to rate perfectly acceptable originals as modified. We leave a more thorough investigation to further work. See similar observations on the effectiveness of untrained ConvNets in vision by Ulyanov et al. (2017).

Probing task comparison A good encoder, such as NMT-trained BiLSTM-max, shows generally good performance across probing tasks. At one extreme, performance is not particularly high on the surface tasks, which might be an indirect sign of the encoder extracting “deeper” linguistic properties. At the other end, performance is still far from the human bounds on TreeDepth, BShift, SOMO and CoordInv. The last three tasks ask if a sentence is syntactically or semantically anomalous. This is a daunting job for an encoder that has not been explicitly trained on acceptability, and it is interesting that the best models are, at least to a certain extent, able to produce reasonable anomaly judgments. The asymmetry between the difficult TreeDepth and easier TopConst is also interesting. Intuitively, TreeDepth requires more nuanced syntactic information (down to the deepest leaf of the tree) than TopConst, that only requires identifying broad chunks.

Figure 3.10 reports how probing task accuracy changes in function of encoder training epochs. The figure shows that NMT probing performance is largely independent of target language, with strikingly similar development patterns across French, German and Finnish. Note in particular the similar probing accuracy curves in French and Finnish, while the corresponding BLEU scores (in lavender) are consistently higher in the former language. For both NMT and SkipThought, WC performance keeps increasing with epochs. For the other tasks, we observe instead an early flattening of the NMT probing curves, while BLEU performance keeps increasing. Most strikingly, SentLen performance is actually *decreasing*, suggesting again that, as a model captures deeper linguistic properties, it will tend to forget about this superficial feature. Finally, for the challenging SOMO task, the curves are mostly flat, suggesting that what BiLSTM-max is able to capture about this task is already encoded in its architecture, and further training doesn’t help much.

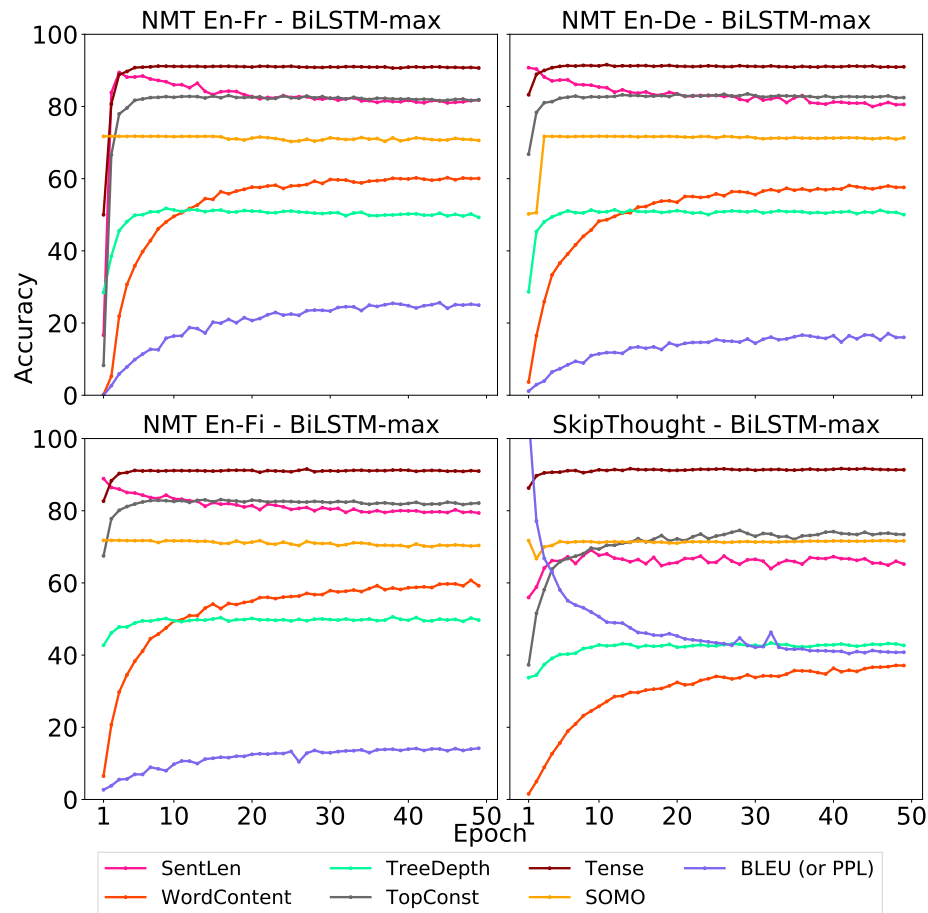


Figure 3.10: **Probing task scores after each training epoch, for NMT and SkipThought.** We also report training score evolution: BLEU for NMT; perplexity (PPL) for SkipThought.

Probing vs. downstream tasks Figure 3.11 reports correlation between performance on our probing tasks and the downstream tasks of SentEval. Strikingly, WC is significantly positively correlated with all downstream tasks. This suggests that, at least for current models, the latter do not require extracting particularly abstract knowledge from the data. Just relying on the *words* contained in the input sentences can get you a long way. Conversely, there is a significant negative correlation between SentLen and most downstream tasks. The number of words in a sentence is not informative about its linguistic contents. The more models abstract away from such information, the more likely it is they will use their capacity to capture more interesting features, as the decrease of the SentLen curve along

training (see Figure 3.10) also suggests. CoordInv and, especially, SOMO, the tasks requiring the most sophisticated semantic knowledge, are those that positively correlate with the largest number of downstream tasks after WC. We observe intriguing asymmetries: SOMO correlates with the SICK-E sentence entailment test, but not with SICK-R, which is about modeling sentence relatedness intuitions. Indeed, logical entailment requires deeper semantic analysis than modeling similarity judgments. TopConst and the number tasks negatively correlate with various similarity and sentiment data sets (SST, STS, SICK-R). This might expose biases in these tasks: SICK-R, for example, deliberately contains sentence pairs with opposite voice, that will have different constituent structure but equal meaning (Marelli et al., 2014). It might also mirror genuine factors affecting similarity judgments (e.g., two sentences differing only in object number are very similar). Remarkably, TREC question type classification is the downstream task correlating with most probing tasks. Question classification is certainly an outlier among our downstream tasks, but we must leave a full understanding of this behaviour to future work (this is exactly the sort of analysis our probing tasks should stimulate).

3.3.5 Related work

Adi et al. (2017) introduced SentLen, WC and a word order test, focusing on a bag-of-vectors baseline, an autoencoder and skip-thought (all trained on the same data used for the probing tasks). We recast their tasks so that they only require a sentence embedding as input (two of their tasks also require word embeddings, polluting sentence-level evaluation), we extend the evaluation to more tasks, encoders and training objectives, and we relate performance on the probing tasks with that on downstream tasks. Shi et al. (2016) also use 3 probing tasks, including Tense and TopConst. It is not clear that they controlled for the same factors we considered (in particular, lexical overlap and sentence length), and they use much smaller training sets, limiting classifier-based evaluation to logistic regression. Moreover, they test a smaller set of models, focusing on machine translation.

Belinkov et al. (2017a), Belinkov et al. (2017b) and Dalvi et al. (2017) are also interested in understanding the type of linguistic knowledge encoded in sentence and word embeddings, but their focus is on word-level morphosyntax and lexical semantics, and

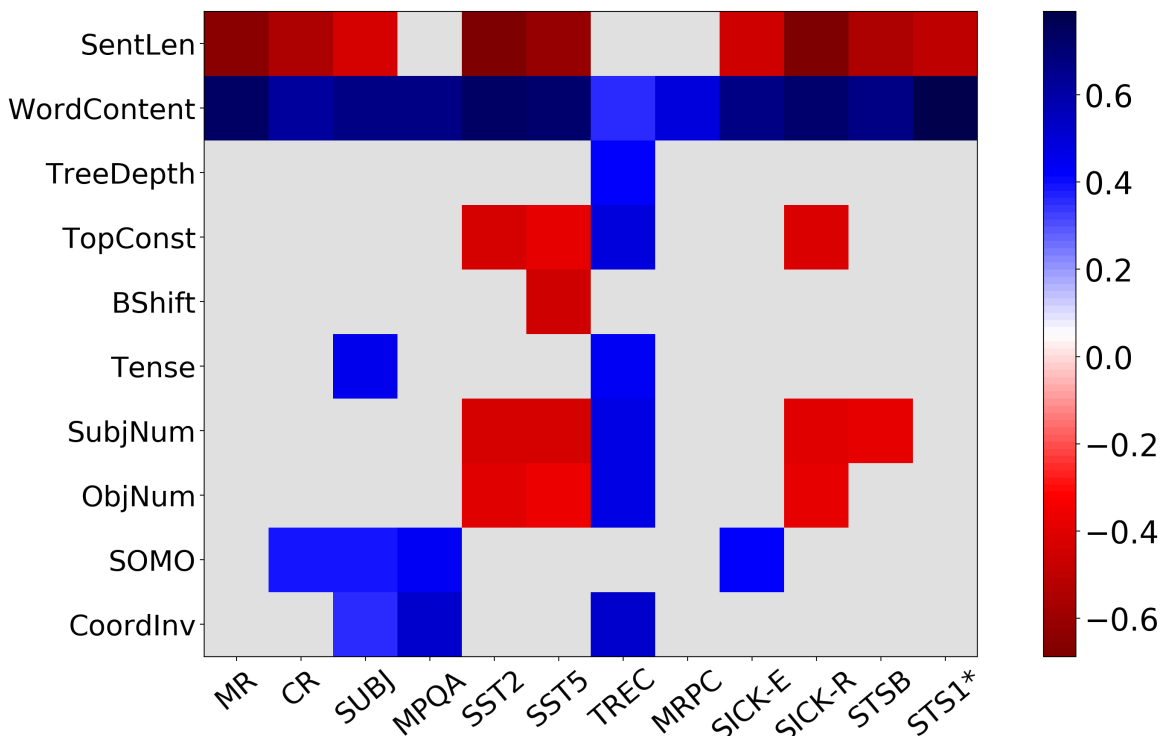


Figure 3.11: **Spearman correlation matrix between probing and downstream tasks.** Correlations based on all sentence embeddings we investigated (more than 40). Cells in gray denote task pairs that are not significantly correlated (after correcting for multiple comparisons).

specifically on NMT encoders and decoders. Sennrich (2017) also focuses on NMT systems, and proposes a contrastive test to assess how they handle various linguistic phenomena. Other work explores the linguistic behaviour of recurrent networks and related models by using visualization, input/hidden representation deletion techniques or by looking at the word-by-word behaviour of the network (e.g., Nagamine et al., 2015; Hupkes et al., 2017; Li et al., 2016; Linzen et al., 2016; Kàdàr et al., 2017; Li et al., 2017). These methods, complementary to ours, are not agnostic to encoder architecture, and cannot be used for general-purpose cross-model evaluation.

3.3.6 Complementary analysis

Amazon Mechanical Turk survey

Subjects were recruited through the standard Amazon Mechanical Turk interface.¹³ We created independent surveys for the SOMO, CoordInv and BShift tasks. We asked subjects to identify which sentences were acceptable and which were anomalous/inverted. Participants were restricted to those based in an English-speaking country.

To maximize annotation quality, we created a control set. Two authors annotated 200 random sentences from each task in a blind pretest. Those sentences on which they agreed were included in the control set.

We collected at least 10 judgments per sentence, for 1k random sentences from each task. We only retained judgments by subjects that rated at least 10 control sentences with accuracy of at least 90%. After filtering, we were left with averages of 2.5, 2.9 and 12 judgments per sentence for SOMO, CoordInv and BShift, respectively. Responses were aggregated by majority voting, before computing the final accuracies.

We did not record any personal data from subjects, and we only used the judgments in aggregated format to produce the estimated human upper bounds reported in our tables.

Further training details

Encoder training For seq2seq tasks, after hyper-parameter tuning, we chose 2-layer LSTM decoders with 512 hidden units. For NLI, we settled on a multi-layer perceptron with 100 hidden units. As is now common in NMT, we apply Byte Pair Encoding (BPE) (Sennrich, 2017) to target sentences only, with 40k codes (see Table 1 in the main text for examples of transformed target sentences). We tune dropout rate and input embedding size, picking 1024 for BiLSTMs and 512 for Gated ConvNets. We use the Adam optimizer for BiLSTMs and SGD with momentum for Gated ConvNets (after Adam gave very poor results). The encoder representation is fed to the decoder at every time step. For model selection on the validation sets, we use BLEU score¹⁴ for NMT and AutoEncoder, perplexity for SkipThought and accuracy for Seq2Tree and NLI.

¹³<https://www.mturk.com/>

¹⁴MOSES multi-bleu.perl script (Koehn et al., 2007)

Table 3.11 reports test set performance of the various architectures on the original training tasks. For NMT and Seq2Tree, we left out two random sets of 10k sentences from the training data for dev and test. The NLI dev and test sets are the ones of SNLI. Observe how results are similar for the three encoders, while, as discussed in the main text, they differ in terms of the linguistic properties their sentence embeddings are capturing. The last row of the table reports BLEU scores for our BiLSTM architecture trained with attention, showing that the architecture is on par with current NMT models, when attention is introduced. For comparison, our attention-based model obtains 37 BLEU score on the standard WMT’14 En-Fr benchmark.

Model	En-Fr	En-De	En-Fi	Seq2Tree	NLI
Gated ConvNet	25.9	17.0	14.2	52.3	83.5
BiLSTM-last	27.3	17.9	14.3	55.2	84.0
BiLSTM-max	27.0	18.0	14.7	53.7	85.3
BiLSTM-Att	39.1	27.2	21.9	58.4	-

Table 3.11: **Test results for training tasks.** Figure of merit is BLEU score for NMT and accuracy for Seq2Tree and NLI.

Probing task training The probing task results reported in the main text are obtained with a MLP that uses the Sigmoid nonlinearity, which we found to perform better than Tanh. We tune the L^2 regularization parameter, the number of hidden states (in [50, 100, 200]) and the dropout rate (in [0, 0.1, 0.2]) on the validation set of each probing task. Only for WC, which has significantly more output classes (1000) than the other tasks, we report Logistic Regression results, since they were consistently better.

Logistic regression results

Logistic regression performance approximates MLP performance (compare Table 3.12 here to Table 2 in the main text). This suggests that most linguistic properties can be extracted with a linear readout of the embeddings. Interestingly, if we focus on a good model-training

combination, such as BiLSTM-max trained on French NMT, the tasks where the improvement from logistic regression to MLP is relatively large ($>3\%$) are those arguably requiring the most nuanced linguistic knowledge (TreeDepth, SOMO, CoordInv).

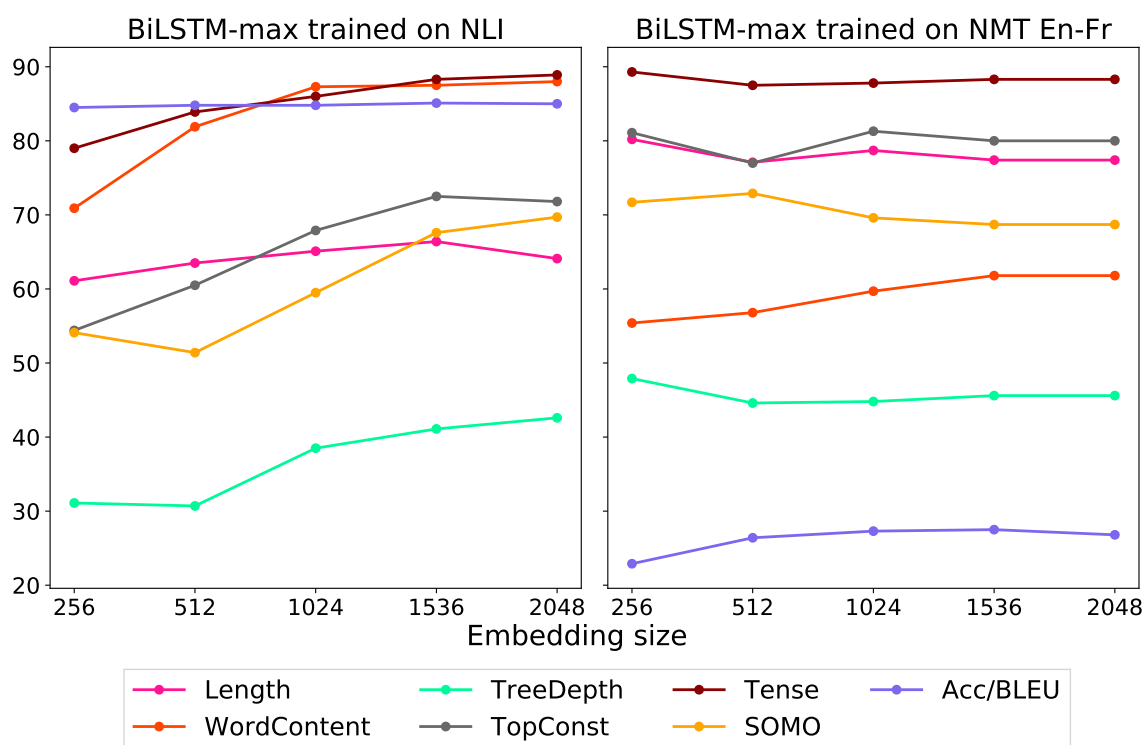


Figure 3.12: **Evolution of probing tasks results wrt. embedding size.** The sentence representations are generated by a BiLSTM-max encoder trained on either NLI or NMT En-Fr, with increasing sentence embedding size.

Downstream task results

We evaluate our architecture+training method combinations on the downstream tasks from the SentEval toolkit.¹⁵ See documentation there for the tasks, that range from subjectivity analysis to question-type classification, to paraphrase detection and entailment. Also refer to the SentEval page and to Conneau et al. (2017) for the specifics of training and figures of merit for each task. In all cases, we used as input our pre-trained embeddings without fine-tuning them to the tasks. Results are reported in Table 3.13.

¹⁵<https://github.com/facebookresearch/SentEval>

We replicate the finding of section 3.2 about the effectiveness of the BiLSTM architecture with max pooling, that has also a slight edge over GatedConvNet (an architecture they did not test). As for the probing tasks, we again notice that BiLSTM-max is already effective without training, and more so than the alternative architectures.

Interestingly, we also confirm Conneau et al.’s finding that NLI is the best source task for pre-training, despite the fact that, as we saw in the main text (Table 2 there), NMT pre-training leads to models that are capturing more linguistic properties. As they observed for downstream tasks, increasing the embedding dimension while adding capacity to the model is beneficial (see Figure 3.12) also for probing tasks in the case of NLI. However, it does not seem to boost the performance of the NMT En-Fr encoder.

Finally, the table also shows results from the literature recently obtained with various state-of-the-art general-purpose encoders, namely: SkipThought with layer normalization (Ba et al., 2016), InferSent (BiLSTM-max as trained on NLI by Conneau et al.) and Multi-Task (Subramanian et al., 2018). A comparison of these results with ours confirms that we are testing models that do not lag much behind the state of the art.

3.3.7 Conclusion

We introduced a set of tasks probing the linguistic knowledge of sentence embedding methods. Their purpose is not to encourage the development of *ad-hoc* models that attain top performance on them, but to help exploring what information is captured by different pre-trained encoders.

We performed an extensive linguistic evaluation of several sentence encoders that were used at the time of this study. Our results suggest that the encoders are capturing a wide range of properties, well above those captured by a set of strong baselines. We further uncovered interesting patterns of correlation between the probing tasks and more complex “downstream” tasks, and presented a set of intriguing findings about the linguistic properties of various embedding methods. For example, we found that Bag-of-Vectors is surprisingly good at capturing sentence-level properties, thanks to redundancies in natural linguistic input. We showed that different encoder architectures trained with the same

objective with similar performance can result in different embeddings, pointing out the importance of the architecture prior for sentence embeddings. In particular, we found that BiLSTM-max embeddings are already capturing interesting linguistic knowledge before training, and that, after training, they detect semantic acceptability without having been exposed to anomalous sentences before. We hope that our publicly available probing task set will become a standard benchmarking tool of the linguistic properties of new encoders, and that it will stir research towards a better understanding of what they learn.

We would like to extend the probing tasks to other languages (which should be relatively easy, given that they are automatically generated), investigate how multi-task training affects probing task performance and leverage our probing tasks to find more linguistically-aware universal encoders.

Task	SentLen	WC	TreeDepth	TopConst	BShift	Tense	SubjNum	ObjNum	SOMO	CoordInv
<i>Baseline representations</i>										
Majority vote	20.0	0.5	17.9	5.0	50.0	50.0	50.0	50.0	50.0	50.0
Hum. Eval.	100	100	84.0	84.0	98.0	85.0	88.0	86.5	81.2	85.0
Length	100	0.2	18.1	9.3	50.6	56.5	50.3	50.1	50.2	50.0
NB-uni-tfidf	22.7	97.8	24.1	41.9	49.5	77.7	68.9	64.0	38.0	50.5
NB-bi-tfidf	23.0	95.0	24.6	53.0	63.8	75.9	69.1	65.4	39.9	55.7
BoV fastText	54.8	91.6	32.3	63.1	50.8	87.8	81.9	79.3	50.3	52.7
<i>BiLSTM-last encoder</i>										
Untrained	32.6	43.8	24.6	74.1	52.2	83.7	82.8	71.8	49.9	64.5
AutoEncoder	98.9	23.3	28.2	72.5	60.1	80.0	81.2	76.8	50.7	62.5
NMT En-Fr	82.9	55.6	35.8	79.8	59.6	86.0	87.6	85.5	50.3	66.1
NMT En-De	82.7	53.1	35.2	80.1	58.3	86.6	88.3	84.5	50.5	66.1
NMT En-Fi	81.7	52.6	35.2	79.3	57.5	86.5	84.4	82.6	50.5	65.9
Seq2Tree	93.2	14.0	46.4	88.5	74.9	87.3	90.5	89.7	50.6	63.4
SkipThought	59.5	35.9	30.2	73.1	58.4	88.7	78.4	76.4	53.0	64.6
NLI	71.6	47.3	28.4	67.4	53.3	77.3	76.6	69.6	51.6	64.7
<i>BiLSTM-max encoder</i>										
Untrained	66.2	88.8	43.1	68.8	70.3	88.7	84.6	81.7	73.0	69.1
AutoEncoder	98.5	17.5	42.3	71.0	69.5	85.7	85.0	80.9	73.0	70.9
NMT En-Fr	79.3	58.3	45.7	80.5	71.2	87.8	88.1	86.3	69.9	71.8
NMT En-De	78.2	56.0	46.9	81.0	69.8	89.1	89.7	87.9	71.3	73.5
NMT En-Fi	77.5	58.3	45.8	80.5	69.7	88.2	88.9	86.1	71.9	72.8
Seq2Tree	91.8	10.3	54.6	88.7	80.0	89.5	91.8	90.7	68.6	69.8
SkipThought	59.6	35.7	42.7	70.5	73.4	90.1	83.3	79.0	70.3	70.1
NLI	65.1	87.3	38.5	67.9	63.8	86.0	78.9	78.5	59.5	64.9
<i>GatedConvNet encoder</i>										
Untrained	90.3	17.1	30.3	47.5	62.0	78.2	72.2	70.9	61.4	59.1
AutoEncoder	99.3	16.8	41.9	69.6	68.1	85.4	85.4	82.1	69.8	70.6
NMT En-Fr	84.3	41.3	36.9	73.8	63.7	85.6	85.7	83.8	58.8	68.1
NMT En-De	87.6	49.0	44.7	78.8	68.8	89.5	89.6	86.8	69.5	70.0
NMT En-Fi	89.1	51.5	44.1	78.6	67.2	88.7	88.5	86.3	68.3	71.0
Seq2Tree	94.5	8.7	53.1	87.4	80.9	89.6	91.5	90.8	68.3	71.6
SkipThought	73.2	48.4	40.4	76.2	71.6	89.8	84.0	79.8	68.9	68.0
NLI	70.9	29.2	38.8	59.3	66.8	80.1	77.7	72.8	69.0	69.1

Table 3.12: **Probing task accuracies with Logistic Regression.** Taking pre-learned sentence embeddings as input.

Model	MR	CR	SUBJ	MPQA	SST-2	SST-5	TREC	MRPC	SICK-E	SICK-R	STSB
<i>Baseline representations</i>											
Chance	50.0	63.8	50.0	68.8	50.0	28.6	21.2	66.5	56.7	0.0	0.0
BoV fastText	78.2	80.2	91.8	88.0	82.3	45.1	83.4	74.4	78.9	82.0	70.2
<i>BiLSTM-last encoder</i>											
Untrained	69.7	70.2	84.8	87.0	77.2	37.6	79.6	68.5	71.6	68.2	54.8
AutoEncoder	66.0	70.7	85.7	81.1	70.0	36.2	84.0	69.9	72.2	67.6	58.3
NMT En-Fr	74.5	78.7	90.3	88.9	79.5	42.0	91.2	73.7	79.7	78.3	69.9
NMT En-De	74.8	78.4	89.8	88.7	78.8	42.3	88.0	74.1	78.8	77.5	69.3
NMT En-Fi	74.2	78.0	89.6	88.9	78.4	39.6	84.6	75.6	79.1	77.1	67.1
Seq2Tree	62.5	69.3	85.7	78.7	64.4	33.0	86.4	73.6	71.9	59.1	44.8
SkipThought	77.1	78.9	92.2	86.7	81.3	43.9	82.4	72.7	77.8	80.0	73.9
NLI	77.3	84.1	88.1	88.6	81.7	43.9	86.0	74.8	83.9	85.6	74.2
<i>BiLSTM-max encoder</i>											
Untrained	75.6	78.2	90.0	88.1	79.9	39.1	80.6	72.2	80.8	83.3	70.2
AutoEncoder	68.3	74.0	87.2	84.6	70.8	34.0	85.0	71.0	75.3	70.4	55.1
NMT En-Fr	76.5	81.1	91.4	89.7	77.7	42.2	89.6	75.1	79.3	78.8	68.8
NMT En-De	77.7	81.2	92.0	89.7	79.3	41.0	88.2	76.2	81.0	80.0	68.7
NMT En-Fi	77.0	81.1	91.5	90.0	80.3	43.4	87.2	75.0	81.7	80.3	69.5
Seq2Tree	65.2	74.4	88.3	80.2	66.5	31.6	85.0	72.0	74.8	65.1	36.1
SkipThought	78.0	82.8	93.0	87.3	81.5	41.9	86.8	73.2	80.0	82.0	71.5
NLI	79.2	86.7	90.0	89.8	83.5	46.4	86.0	74.5	84.5	87.5	76.6
<i>GatedConvNet encoder</i>											
Untrained	65.5	65.3	78.3	82.9	65.8	34.0	67.6	68.1	61.6	56.7	38.9
AutoEncoder	72.1	74.1	86.6	86.0	74.4	36.6	79.6	69.7	72.0	65.8	45.5
NMT En-Fr	74.5	78.3	88.7	88.4	76.8	38.3	86.2	72.5	77.3	73.2	60.4
NMT En-De	77.1	80.4	90.9	89.2	79.2	41.9	90.4	76.8	81.9	78.7	69.4
NMT En-Fi	76.9	82.0	91.2	90.0	78.8	41.9	90.0	76.7	81.1	79.5	70.8
Seq2Tree	65.3	73.1	85.0	79.8	63.7	31.8	81.2	72.9	74.0	58.4	30.8
SkipThought	76.0	81.7	91.5	87.2	77.9	41.5	88.8	72.3	79.5	80.0	67.8
NLI	76.7	84.7	87.4	89.1	79.2	40.9	82.0	70.8	82.0	84.7	64.4
<i>Other sentence embedding methods</i>											
SkipThought	79.4	83.1	93.7	89.3	82.9	-	88.4	72.4	79.5	85.8	72.1
InferSent	81.1	86.3	92.4	90.2	84.6	46.3	88.2	76.2	86.3	88.4	75.8
MultiTask	82.4	88.6	93.8	90.7	85.1	-	94.0	78.3	87.4	88.5	78.7

Table 3.13: Downstream tasks results for various sentence encoder architectures pre-trained in different ways.

Chapter Summary

In this chapter, we have explained how to evaluate, learn and analyze universal fixed-size sentence embedding spaces, through SentEval, InferSent and probing tasks (Conneau and Kiela, 2018; Conneau et al., 2017, 2018a). After our work on universal sentence representations, several extensions have been made by the community. In particular, multi-task training (Subramanian et al., 2018) was used to build even better general-purpose sentence encoder with the intuition that if sentence encoders are to be universal, they should be trained on multiple tasks at once. In parallel, another similar line of work appeared, dealing with language model pretraining. Instead of using supervised data, (Radford et al., 2018; Devlin et al., 2018) revisited the power of language modeling for learning generic representations of text. Their approach was different than ours in several way. They fine-tune the entire encoder on the downstream task while we only learn a classifier on top of frozen fixed-size sentence embeddings. And they evaluate their model on the GLUE benchmark, which is different than SentEval as it includes high-resource downstream tasks while SentEval is mostly focused on low-resource transfer. However they showed that language model fine-tuning also worked well when considering a limited amount of data from the downstream tasks. In our contribution in section 4.3, we present in further details that line of work as part of our own contributions on cross-lingual language modeling. After our contributions on fixed-size sentence representations, we continued working on learning text representations but in the cross-lingual setting. I got interested in aligning distributions of words or sentences (Conneau et al., 2018b,c; Lample and Conneau, 2019). There are multiple reasons why cross-lingual understanding is an interesting direction for research. There are approximately 7000 languages in the world, but a large amount of the available supervised (or unsupervised) language data is in a few high-resource languages. Building annotated data in each language is simply not a scalable way to provide strong NLP systems in all languages. Ideally, we would like to leverage the supervised data that exists in one high-resource language such as English to build or improve an NLP system in a low-resource language like Swahili. Even if our methods only required unsupervised text data like BERT, the amount of text from a language like Nepalese is simply not sufficient to build strong representations and we may want to leverage weak supervision from

other languages, even in the form of unsupervised text. Also, having one single system for each language is cumbersome and hard to maintain. In the next chapter, we explain how we can align distributions of words and sentences with as little parallel supervision as possible. Similar to SentEval, we build a zero-shot cross-lingual classification benchmark called "XNLI" to evaluate cross-lingual encoders. We show that with cross-lingual encoders (Lample and Conneau, 2019), we can leverage English annotated data to build stronger classification systems in Urdu or Swahili.

Chapter 4

Cross-lingual sentence representations

In this chapter, we first present how to align word embedding spaces without parallel data which is the premise of a new machine translation paradigm called "unsupervised machine translation". We then switch to the sentence-level and explain how and why we built the first large-scale cross-lingual understanding (XLU) evaluation benchmark for zero-shot text classification. We use this new benchmark called XNLI to evaluate a sentence embedding alignment method based on an alignment loss that leverages available parallel data. While having seen only supervision in English at training time, we are able to make very accurate predictions at test time for the fourteen XNLI languages. Our approach can have an important impact in a production setting as supervised data is usually scarce in many languages. One problem of that approach is that it requires a significant amount of parallel data which is by definition difficult to collect for low-resource languages. That is why, inspired by recent advances in language model fine-tuning, we investigate cross-lingual language models as a way to learn powerful cross-lingual encoders without parallel data. Similar to the unsupervised alignment of word embeddings, we show that we can align sentence embedding spaces in a completely unsupervised way. We extend the BERT approach to the cross-lingual setting and also propose a new objective to leverage parallel data when it is available. Our approach leads to a new state of the art on XNLI and significantly outperforms previous approaches. I also demonstrate in the last section that cross-lingual language model pretraining of encoders and decoders in the case of supervised and unsupervised neural machine translation significantly outperforms the previous state of the

art.

4.1 Aligning Word Representations Without Parallel Data

State-of-the-art methods for learning cross-lingual word embeddings have relied on bilingual dictionaries or parallel corpora. Recent studies showed that the need for parallel data supervision can be alleviated with character-level information. While these methods showed encouraging results, they are not on par with their supervised counterparts and are limited to pairs of languages sharing a common alphabet. In this section, we show that we can build a bilingual dictionary between two languages without using any parallel corpora, by aligning monolingual word embedding spaces in an unsupervised way. Without using any character information, our model even outperforms existing supervised methods on cross-lingual tasks for some language pairs. Our experiments demonstrate that our method works very well also for distant language pairs, like English-Russian or English-Chinese. We finally describe experiments on the English-Esperanto low-resource language pair, on which there only exists a limited amount of parallel data, to show the potential impact of our method in fully unsupervised machine translation. Our code, embeddings and dictionaries are publicly available¹.

4.1.1 Introduction

Most successful methods for learning distributed representations of words (e.g. (Mikolov et al., 2013c,a; Pennington et al., 2014; Bojanowski et al., 2017)) rely on the distributional hypothesis of (Harris, 1954), which states that words occurring in similar contexts tend to have similar meanings. (Levy and Goldberg, 2014) show that the skip-gram with negative sampling method of (Mikolov et al., 2013c) amounts to factorizing a word-context co-occurrence matrix, whose entries are the pointwise mutual information of the respective word and context pairs. Exploiting word co-occurrence statistics leads to word vectors that reflect the semantic similarities and dissimilarities: similar words are close in the embedding space and conversely.

¹<https://github.com/facebookresearch/MUSE>

(Mikolov et al., 2013b) first noticed that continuous word embedding spaces exhibit similar structures across languages, even when considering distant language pairs like English and Vietnamese. They proposed to exploit this similarity by learning a linear mapping from a source to a target embedding space. They employed a parallel vocabulary of five thousand words as anchor points to learn this mapping and evaluated their approach on a word translation task. Since then, several studies aimed at improving these cross-lingual word embeddings ((Faruqui and Dyer, 2014; Xing et al., 2015; Lazaridou et al., 2015; Ammar et al., 2016; Artetxe et al., 2016; Smith et al., 2017)), but they all rely on bilingual word lexicons.

Previous attempts at reducing the need for bilingual supervision (Smith et al., 2017) employ identical character strings to form a parallel vocabulary. The iterative method of Artetxe et al. (2017) gradually aligns embedding spaces, starting from a parallel vocabulary of aligned digits. These methods are however limited to similar languages sharing a common alphabet, such as European languages. Some recent methods explored distribution-based approach (Cao et al., 2016) or adversarial training (Zhang et al., 2017a) to obtain cross-lingual word embeddings without any parallel data. While these approaches sound appealing, their performance is significantly below supervised methods. To sum up, current methods have either not reached competitive performance, or they still require parallel data, such as aligned corpora (Gouws et al., 2015; Vulic and Moens, 2015) or a seed parallel lexicon (Duong et al., 2016).

In this section, we introduce a model that either is on par, or outperforms supervised state-of-the-art methods, without employing any cross-lingual annotated data. We only use two large monolingual corpora, one in the source and one in the target language. Our method leverages adversarial training to learn a linear mapping from a source to a target space and operates in two steps. First, in a two-player game, a discriminator is trained to distinguish between the mapped source embeddings and the target embeddings, while the mapping (which can be seen as a generator) is jointly trained to fool the discriminator. Second, we extract a synthetic dictionary from the resulting shared embedding space and fine-tune the mapping with the closed-form Procrustes solution from (Schönemann, 1966). Since the method is unsupervised, cross-lingual data can not be used to select the best model. To overcome this issue, we introduce an unsupervised selection metric that is highly

correlated with the mapping quality and that we use both as a stopping criterion and to select the best hyper-parameters.

In summary, this section makes the following main contributions:

- We present an unsupervised approach that reaches or outperforms state-of-the-art supervised approaches on several language pairs and on three different evaluation tasks, namely word translation, sentence translation retrieval, and cross-lingual word similarity. On a standard word translation retrieval benchmark, using 200k vocabularies, our method reaches 66.2% accuracy on English-Italian while the best supervised approach is at 63.7%.
- We introduce a cross-domain similarity adaptation to mitigate the so-called hubness problem (points tending to be nearest neighbors of many points in high-dimensional spaces). It is inspired by the self-tuning method from (Zelnik-manor and Perona, 2005), but adapted to our two-domain scenario in which we must consider a bi-partite graph for neighbors. This approach significantly improves the absolute performance, and outperforms the state of the art both in supervised and unsupervised setups on word-translation benchmarks.
- We propose an unsupervised criterion that is highly correlated with the quality of the mapping, that can be used both as a stopping criterion and to select the best hyper-parameters.
- We release high-quality dictionaries for 12 oriented languages pairs, as well as the corresponding supervised and unsupervised word embeddings.
- We demonstrate the effectiveness of our method using an example of a low-resource language pair where parallel corpora are not available (English-Esperanto) for which our method is particularly suited.

The section is organized as follows. Section 4.1.2 describes our unsupervised approach with adversarial training and our refinement procedure. We then present our training procedure with unsupervised model selection in Section 4.1.3. We report in Section 4.1.4 our results on several cross-lingual tasks for several language pairs and compare our approach

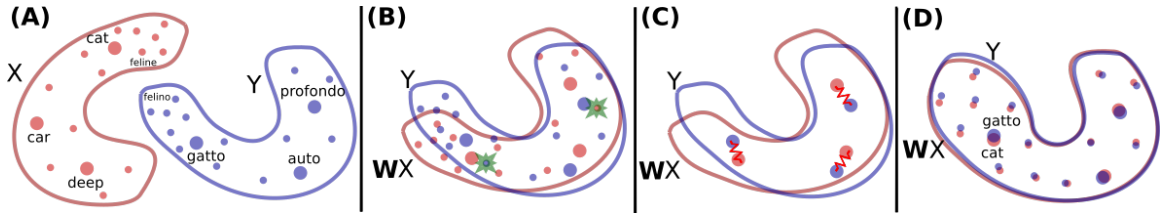


Figure 4.1: **Toy illustration of the method.** (A) There are two distributions of word embeddings, English words in red denoted by X and Italian words in blue denoted by Y , which we want to align/translate. Each dot represents a word in that space. The size of the dot is proportional to the frequency of the words in the training corpus of that language. (B) Using adversarial learning, we learn a rotation matrix W which roughly aligns the two distributions. The green stars are randomly selected words that are fed to the discriminator to determine whether the two word embeddings come from the same distribution. (C) The mapping W is further refined via Procrustes. This method uses frequent words aligned by the previous step as anchor points, and minimizes an energy function that corresponds to a spring system between anchor points. The refined mapping is then used to map all words in the dictionary. (D) Finally, we translate by using the mapping W and a distance metric, dubbed CSLS, that expands the space where there is high density of points (like the area around the word “cat”), so that “hubs” (like the word “cat”) become less close to other word vectors than they would otherwise (compare to the same region in panel (A)).

to supervised methods. Finally, we explain how our approach differs from recent related work on learning cross-lingual word embeddings.

4.1.2 Model

In this section, we always assume that we have two sets of embeddings trained independently on monolingual data. Our work focuses on learning a mapping between the two sets such that translations are close in the shared space. (Mikolov et al., 2013b) show that they can exploit the similarities of monolingual embedding spaces to learn such a mapping. For this purpose, they use a known dictionary of $n = 5000$ pairs of words $\{x_i, y_i\}_{i \in \{1, n\}}$, and learn a linear mapping W between the source and the target space such that

$$W^* = \underset{W \in M_d(\mathbb{R})}{\operatorname{argmin}} \|WX - Y\|_F \quad (4.1)$$

where d is the dimension of the embeddings, $M_d(\mathbb{R})$ is the space of $d \times d$ matrices of real numbers, and X and Y are two aligned matrices of size $d \times n$ containing the embeddings of the words in the parallel vocabulary. The translation t of any source word s is defined as $t = \operatorname{argmax}_t \cos(Wx_s, y_t)$. In practice, (Mikolov et al., 2013b) obtained better results on the word translation task using a simple linear mapping, and did not observe any improvement when using more advanced strategies like multilayer neural networks. (Xing et al., 2015) showed that these results are improved by enforcing an orthogonality constraint on W . In that case, the equation (4.1) boils down to the Procrustes problem, which advantageously offers a closed form solution obtained from the singular value decomposition (SVD) of YX^T :

$$W^* = \operatorname{argmin}_{W \in O_d(\mathbb{R})} \|WX - Y\|_F = UV^T, \text{ with } U\Sigma V^T = \operatorname{SVD}(YX^T). \quad (4.2)$$

In this section, we show how to learn this mapping W without cross-lingual supervision; an illustration of the approach is given in Fig. 4.1. First, we learn an initial proxy of W by using an adversarial criterion. Then, we use the words that match the best as anchor points for Procrustes. Finally, we improve performance over less frequent words by changing the metric of the space, which leads to spread more of those points in dense regions. Next, we describe the details of each of these steps.

Domain-adversarial setting

In this section, we present our domain-adversarial approach for learning W without cross-lingual supervision. Let $\mathcal{X} = \{x_1, \dots, x_n\}$ and $\mathcal{Y} = \{y_1, \dots, y_m\}$ be two sets of n and m word embeddings coming from a source and a target language respectively. A model is trained to discriminate between elements randomly sampled from $W\mathcal{X} = \{Wx_1, \dots, Wx_n\}$ and \mathcal{Y} . We call this model the discriminator. W is trained to prevent the discriminator from making accurate predictions. As a result, this is a two-player game, where the discriminator aims at maximizing its ability to identify the origin of an embedding, and W aims at preventing the discriminator from doing so by making $W\mathcal{X}$ and \mathcal{Y} as *similar* as possible. This approach is in line with the work of (Ganin et al., 2016), who proposed to learn latent representations invariant to the input domain, where in our case, a domain is represented by a language (source or target).

Discriminator objective We refer to the discriminator parameters as θ_D . We consider the probability $P_{\theta_D}(\text{source} = 1|z)$ that a vector z is the mapping of a source embedding (as opposed to a target embedding) according to the discriminator. The discriminator loss can be written as:

$$\mathcal{L}_D(\theta_D|W) = -\frac{1}{n} \sum_{i=1}^n \log P_{\theta_D}(\text{source} = 1|Wx_i) - \frac{1}{m} \sum_{i=1}^m \log P_{\theta_D}(\text{source} = 0|y_i). \quad (4.3)$$

Mapping objective In the unsupervised setting, W is now trained so that the discriminator is unable to accurately predict the embedding origins:

$$\mathcal{L}_W(W|\theta_D) = -\frac{1}{n} \sum_{i=1}^n \log P_{\theta_D}(\text{source} = 0|Wx_i) - \frac{1}{m} \sum_{i=1}^m \log P_{\theta_D}(\text{source} = 1|y_i). \quad (4.4)$$

Learning algorithm To train our model, we follow the standard training procedure of deep adversarial networks of (Goodfellow et al., 2014). For every input sample, the discriminator and the mapping matrix W are trained successively with stochastic gradient updates to respectively minimize \mathcal{L}_D and \mathcal{L}_W . The details of training are given in the next section.

Refinement procedure

The matrix W obtained with adversarial training gives good performance (see Table 4.1), but the results are still not on par with the supervised approach. In fact, the adversarial approach tries to align all words irrespective of their frequencies. However, rare words have embeddings that are less updated and are more likely to appear in different contexts in each corpus, which makes them harder to align. Under the assumption that the mapping is linear, it is then better to infer the global mapping using only the most frequent words as anchors. Besides, the accuracy on the most frequent word pairs is high after adversarial training.

To refine our mapping, we build a synthetic parallel vocabulary using the W just learned with adversarial training. Specifically, we consider the most frequent words and retain only mutual nearest neighbors to ensure a high-quality dictionary. Subsequently, we apply the

Procrustes solution in (4.2) on this generated dictionary. Considering the improved solution generated with the Procrustes algorithm, it is possible to generate a more accurate dictionary and apply this method iteratively, similarly to (Artetxe et al., 2017). However, given that the synthetic dictionary obtained using adversarial training is already strong, we only observe small improvements when doing more than one iteration, i.e., the improvements on the word translation task are usually below 1%.

Cross-domain similarity local scaling (CSLS)

In this subsection, our motivation is to produce reliable matching pairs between two languages: we want to improve the comparison metric such that the nearest neighbor of a source word, in the target language, is more likely to have as a nearest neighbor this particular source word.

Nearest neighbors are by nature asymmetric: y being a K -NN of x does not imply that x is a K -NN of y . In high-dimensional spaces (Radovanović et al., 2010), this leads to a phenomenon that is detrimental to matching pairs based on a nearest neighbor rule: some vectors, dubbed *hubs*, are with high probability nearest neighbors of many other points, while others (anti-hubs) are not nearest neighbors of any point. This problem has been observed in different areas, from matching image features in vision (Jegou et al., 2010) to translating words in text understanding applications (Dinu et al., 2015). Various solutions have been proposed to mitigate this issue, some being reminiscent of pre-processing already existing in spectral clustering algorithms (Zelnik-manor and Perona, 2005).

However, most studies aiming at mitigating hubness consider a single feature distribution. In our case, we have two domains, one for each language. This particular case is taken into account by (Dinu et al., 2015), who propose a pairing rule based on reverse ranks, and the inverted soft-max (ISF) by (Smith et al., 2017), which we evaluate in our experimental section. These methods are not fully satisfactory because the similarity updates are different for the words of the source and target languages. Additionally, ISF requires to cross-validate a parameter, whose estimation is noisy in an unsupervised setting where we do not have a direct cross-validation criterion.

In contrast, we consider a bi-partite neighborhood graph, in which each word of a given dictionary is connected to its K nearest neighbors in the other language. We denote by

$\mathcal{N}_T(Wx_s)$ the neighborhood, on this bi-partite graph, associated with a mapped source word embedding Wx_s . All K elements of $\mathcal{N}_T(Wx_s)$ are words from the target language. Similarly we denote by $\mathcal{N}_S(y_t)$ the neighborhood associated with a word t of the target language. We consider the mean similarity of a source embedding x_s to its target neighborhood as

$$r_T(Wx_s) = \frac{1}{K} \sum_{y_t \in \mathcal{N}_T(Wx_s)} \cos(Wx_s, y_t), \quad (4.5)$$

where $\cos(\cdot, \cdot)$ is the cosine similarity. Likewise we denote by $r_S(y_t)$ the mean similarity of a target word y_t to its neighborhood. These quantities are computed for all source and target word vectors with the efficient nearest neighbors implementation by (Johnson et al., 2017a). We use them to define a similarity measure $\text{CSLS}(\cdot, \cdot)$ between mapped source words and target words, as

$$\text{CSLS}(Wx_s, y_t) = 2 \cos(Wx_s, y_t) - r_T(Wx_s) - r_S(y_t). \quad (4.6)$$

Intuitively, this update increases the similarity associated with isolated word vectors. Conversely it decreases the ones of vectors lying in dense areas. Our experiments show that the CSLS significantly increases the accuracy for word translation retrieval, while not requiring any parameter tuning.

4.1.3 Training and architectural choices

Architecture

We use unsupervised word vectors that were trained using fastText². These correspond to monolingual embeddings of dimension 300 trained on Wikipedia corpora; therefore, the mapping W has size 300×300 . Words are lower-cased, and those that appear less than 5 times are discarded for training. As a post-processing step, we only select the first 200k most frequent words in our experiments.

For our discriminator, we use a multilayer perceptron with two hidden layers of size 2048, and Leaky-ReLU activation functions. The input to the discriminator is corrupted

²Word vectors downloaded from: <https://github.com/facebookresearch/fastText>

with dropout noise with a rate of 0.1. As suggested by (Goodfellow, 2016), we include a smoothing coefficient $s = 0.2$ in the discriminator predictions. We use stochastic gradient descent with a batch size of 32, a learning rate of 0.1 and a decay of 0.95 both for the discriminator and W . We divide the learning rate by 2 every time our unsupervised validation criterion decreases.

Discriminator inputs

The embedding quality of rare words is generally not as good as the one of frequent words (Luong et al., 2013), and we observed that feeding the discriminator with rare words had a small, but not negligible negative impact. As a result, we only feed the discriminator with the 50,000 most frequent words. At each training step, the word embeddings given to the discriminator are sampled uniformly. Sampling them according to the word frequency did not have any noticeable impact on the results.

Orthogonality

(Smith et al., 2017) showed that imposing an orthogonal constraint to the linear operator led to better performance. Using an orthogonal matrix has several advantages. First, it ensures that the monolingual quality of the embeddings is preserved. Indeed, an orthogonal matrix preserves the dot product of vectors, as well as their ℓ_2 distances, and is therefore an isometry of the Euclidean space (such as a rotation). Moreover, it made the training procedure more stable in our experiments. In this section, we propose to use a simple update step to ensure that the matrix W stays close to an orthogonal matrix during training ((Cisse et al., 2017)). Specifically, we alternate the update of our model with the following update rule on the matrix W :

$$W \leftarrow (1 + \beta)W - \beta(WW^T)W \quad (4.7)$$

where $\beta = 0.01$ is usually found to perform well. This method ensures that the matrix stays close to the manifold of orthogonal matrices after each update. In practice, we observe that the eigenvalues of our matrices all have a modulus close to 1, as expected.

Dictionary generation

The refinement step requires to generate a new dictionary at each iteration. In order for the Procrustes solution to work well, it is best to apply it on correct word pairs. As a result, we use the CSLS method described in Section 4.1.2 to select more accurate translation pairs in the dictionary. To increase even more the quality of the dictionary, and ensure that W is learned from correct translation pairs, we only consider mutual nearest neighbors, i.e. pairs of words that are mutually nearest neighbors of each other according to CSLS. This significantly decreases the size of the generated dictionary, but improves its accuracy, as well as the overall performance.

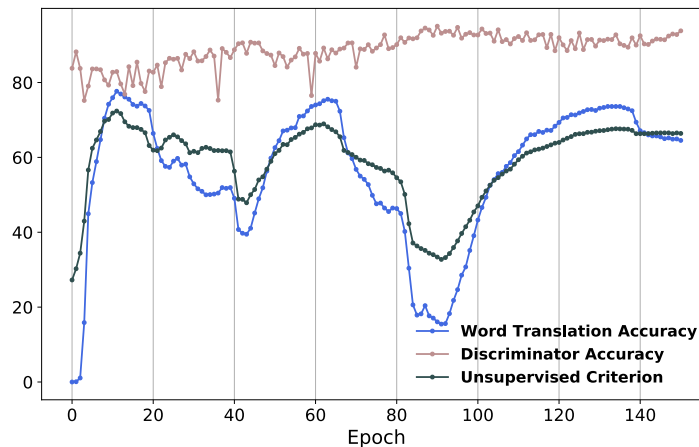


Figure 4.2: **Unsupervised model selection.** Correlation between our unsupervised validation criterion (black line) and actual word translation accuracy (blue line). In this particular experiment, the selected model is at epoch 10. Observe how our criterion is well correlated with translation accuracy.

Validation criterion for unsupervised model selection

Selecting the best model is a challenging, yet important task in the unsupervised setting, as it is not possible to use a validation set (using a validation set would mean that we possess parallel data). To address this issue, we perform model selection using an unsupervised criterion that quantifies the closeness of the source and target embedding spaces. Specifically, we consider the 10k most frequent source words, and use CSLS to generate a translation

for each of them. We then compute the average cosine similarity between these deemed translations, and use this average as a validation metric. We found that this simple criterion is better correlated with the performance on the evaluation tasks than optimal transport distances such as the Wasserstein distance ((Rubner et al., 2000)). Figure 4.2 shows the correlation between the evaluation score and this unsupervised criterion (without stabilization by learning rate shrinkage). We use it as a stopping criterion during training, and also for hyper-parameter selection in all our experiments.

4.1.4 Experiments

In this section, we empirically demonstrate the effectiveness of our unsupervised approach on several benchmarks, and compare it with state-of-the-art supervised methods. We first present the cross-lingual evaluation tasks that we consider to evaluate the quality of our cross-lingual word embeddings. Then, we present our baseline model. Last, we compare our unsupervised approach to our baseline and to previous methods. In the last subsection, we offer a complementary analysis on the alignment of several sets of English embeddings trained with different methods and corpora.

	en-es	es-en	en-fr	fr-en	en-de	de-en	en-ru	ru-en	en-zh	zh-en	en-eo	eo-en
<i>Methods with cross-lingual supervision and fastText embeddings</i>												
Procrustes - NN	77.4	77.3	74.9	76.1	68.4	67.7	47.0	58.2	40.6	30.2	22.1	20.4
Procrustes - ISF	81.1	82.6	81.1	81.3	71.1	71.5	49.5	63.8	35.7	37.5	29.0	27.9
Procrustes - CSLS	81.4	82.9	81.1	82.4	73.5	72.4	51.7	63.7	42.7	36.7	29.3	25.3
<i>Methods without cross-lingual supervision and fastText embeddings</i>												
Adv - NN	69.8	71.3	70.4	61.9	63.1	59.6	29.1	41.5	18.5	22.3	13.5	12.1
Adv - CSLS	75.7	79.7	77.8	71.2	70.1	66.4	37.2	48.1	23.4	28.3	18.6	16.6
Adv - Refine - NN	79.1	78.1	78.1	78.2	71.3	69.6	37.3	54.3	30.9	21.9	20.7	20.6
Adv - Refine - CSLS	81.7	83.3	82.3	82.1	74.0	72.2	44.0	59.1	32.5	31.4	28.2	25.6

Table 4.1: **Word translation retrieval P@1 for our released vocabularies in various language pairs.** We consider 1,500 source test queries, and 200k target words for each language pair. We use fastText embeddings trained on Wikipedia. NN: nearest neighbors. ISF: inverted softmax. ('en' is English, 'fr' is French, 'de' is German, 'ru' is Russian, 'zh' is classical Chinese and 'eo' is Esperanto)

Evaluation tasks

Word translation The task considers the problem of retrieving the translation of given source words. The problem with most available bilingual dictionaries is that they are generated using online tools like Google Translate, and do not take into account the polysemy of words. Failing to capture word polysemy in the vocabulary leads to a wrong evaluation of the quality of the word embedding space. Other dictionaries are generated using phrase tables of machine translation systems, but they are very noisy or trained on relatively small parallel corpora. For this task, we create high-quality dictionaries of up to 100k pairs of words using an internal machine translation tool to alleviate this issue. We make these dictionaries publicly available as part of the MUSE library³.

We report results on these bilingual dictionaries, as well on those released by (Dinu et al., 2015) to allow for a direct comparison with previous approaches. For each language pair, we consider 1,500 query source and 200k target words. Following standard practice, we measure how many times one of the correct translations of a source word is retrieved, and report $\text{precision}@k$ for $k = 1, 5, 10$.

Cross-lingual semantic word similarity We also evaluate the quality of our cross-lingual word embeddings space using word similarity tasks. This task aims at evaluating how well the cosine similarity between two words of different languages correlates with a human-labeled score. We use the SemEval 2017 competition data (Camacho-Collados et al., 2017) which provides large, high-quality and well-balanced datasets composed of nominal pairs that are manually scored according to a well-defined similarity scale. We report Pearson correlation.

Sentence translation retrieval Going from the word to the sentence level, we consider bag-of-words aggregation methods to perform sentence retrieval on the Europarl corpus. We consider 2,000 source sentence queries and 200k target sentences for each language pair and report the $\text{precision}@k$ for $k = 1, 5, 10$, which accounts for the fraction of pairs for which the correct translation of the source words is in the k -th nearest neighbors. We

³<https://github.com/facebookresearch/MUSE>

use the idf-weighted average to merge word into sentence embeddings. The idf weights are obtained using other 300k sentences from Europarl.

Results and discussion

In what follows, we present the results on word translation retrieval using our bilingual dictionaries in Table 4.1 and our comparison to previous work in Table 4.2 where we significantly outperform previous approaches. We also present results on the sentence translation retrieval task in Table 4.3 and the cross-lingual word similarity task in Table 4.4. Finally, we present results on word-by-word translation for English-Esperanto in Table 4.5.

	English to Italian			Italian to English		
	P@1	P@5	P@10	P@1	P@5	P@10
<i>Methods with cross-lingual supervision (WaCky)</i>						
(Mikolov et al., 2013b) †	33.8	48.3	53.9	24.9	41.0	47.4
(Dinu et al., 2015) †	38.5	56.4	63.9	24.6	45.4	54.1
CCA †	36.1	52.7	58.1	31.0	49.9	57.0
(Artetxe et al., 2017)	39.7	54.7	60.5	33.8	52.4	59.1
(Smith et al., 2017) †	43.1	60.7	66.4	38.0	58.5	63.6
Procrustes - CSLS	44.9	61.8	66.6	38.5	57.2	63.0
<i>Methods without cross-lingual supervision (WaCky)</i>						
Adv - Refine - CSLS	45.1	60.7	65.1	38.3	57.8	62.8
<i>Methods with cross-lingual supervision (Wiki)</i>						
Procrustes - CSLS	63.7	78.6	81.1	56.3	76.2	80.6
<i>Methods without cross-lingual supervision (Wiki)</i>						
Adv - Refine - CSLS	66.2	80.4	83.4	58.7	76.5	80.9

Table 4.2: **English-Italian word translation** average precisions (@1, @5, @10) from 1.5k source word queries using 200k target words. Results marked with the symbol † are from (Smith et al., 2017). Wiki means the embeddings were trained on Wikipedia using fastText. Note that the method used by (Artetxe et al., 2017) does not use the same supervision as other supervised methods, as they only use numbers in their initial parallel dictionary.

Baselines In our experiments, we consider a supervised baseline that uses the solution of the Procrustes formula given in (4.2), and trained on a dictionary of 5,000 source words. This baseline can be combined with different similarity measures: NN for nearest neighbor similarity, ISF for Inverted SoftMax and the CSLS approach described in Section 4.1.2.

Cross-domain similarity local scaling This approach has a single parameter K defining the size of the neighborhood. The performance is very stable and therefore K does not need cross-validation: the results are essentially the same for $K = 5, 10$ and 50 , therefore we set $K = 10$ in all experiments. In Table 4.1, we observe the impact of the similarity metric with the Procrustes supervised approach. Looking at the difference between *Procrustes-NN* and *Procrustes-CSLS*, one can see that CSLS provides a strong and robust gain in performance across all language pairs, with up to 7.2% in en-eo. We observe that *Procrustes-CSLS* is almost systematically better than *Procrustes-ISF*, while being computationally faster and not requiring hyper-parameter tuning. In Table 4.2, we compare our *Procrustes-CSLS* approach to previous models presented in (Mikolov et al., 2013b; Dinu et al., 2015; Smith et al., 2017; Artetxe et al., 2017) on the English-Italian word translation task, on which state-of-the-art models have been already compared. We show that our *Procrustes-CSLS* approach obtains an accuracy of 44.9%, outperforming previous approaches. In Table 4.3, we also obtain a strong gain in accuracy in the Italian-English sentence retrieval task using CSLS, from 53.5% to 69.5%, outperforming previous approaches by an absolute gain of more than 20%.

Impact of the monolingual embeddings For the word translation task, we obtained a significant boost in performance when considering fastText embeddings trained on Wikipedia, as opposed to previously used CBOW embeddings trained on the WaCky datasets (Baroni et al., 2009), as can be seen in Table 4.2. Among the two factors of variation, we noticed that this boost in performance was mostly due to the change in corpora. The fastText embeddings, which incorporates more syntactic information about the words, obtained only two percent more accuracy compared to CBOW embeddings trained on the same corpus, out of the 18.8% gain. We hypothesize that this gain is due to the similar co-occurrence statistics of Wikipedia corpora. Figure 4.3 in the appendix shows results on the alignment of different monolingual embeddings and concurs with this hypothesis. We also obtained better results for monolingual evaluation tasks such as word similarities and word analogies when training our embeddings on the Wikipedia corpora.

Adversarial approach Table 4.1 shows that the adversarial approach provides a strong system for learning cross-lingual embeddings without parallel data. On the es-en and en-fr language pairs, *Adv-CSLS* obtains a P@1 of 79.7% and 77.8%, which is only 3.2% and 3.3% below the supervised approach. Additionally, we observe that most systems still obtain decent results on distant languages that do not share a common alphabet (en-ru and en-zh), for which method exploiting identical character strings are just not applicable (Artetxe et al., 2017). This method allows us to build a strong synthetic vocabulary using similarities obtained with CSLS. The gain in absolute accuracy observed with CSLS on the Procrustes method is even more important here, with differences between *Adv-NN* and *Adv-CSLS* of up to 8.4% on es-en. As a simple baseline, we tried to match the first two moments of the projected source and target embeddings, which amounts to solving $W^* \in \operatorname{argmin}_W \|(WX)^T(WX) - Y^TY\|_F$ and solving the sign ambiguity (Umeyama, 1988). This attempt was not successful, which we explain by the fact that this method tries to align only the first two moments, while adversarial training matches all the moments and can learn to focus on specific areas of the distributions instead of considering global statistics.

	English to Italian			Italian to English		
	P@1	P@5	P@10	P@1	P@5	P@10
<i>Methods with cross-lingual supervision</i>						
(Mikolov et al., 2013b) †	10.5	18.7	22.8	12.0	22.1	26.7
(Dinu et al., 2015) †	45.3	72.4	80.7	48.9	71.3	78.3
(Smith et al., 2017) †	54.6	72.7	78.2	42.9	62.2	69.2
Procrustes - NN	42.6	54.7	59.0	53.5	65.5	69.5
Procrustes - CSLS	66.1	77.1	80.7	69.5	79.6	83.5
<i>Methods without cross-lingual supervision</i>						
Adv - CSLS	42.5	57.6	63.6	47.0	62.1	67.8
Adv - Refine - CSLS	65.9	79.7	83.1	69.0	79.7	83.1

Table 4.3: **English-Italian sentence translation retrieval.** We report the average P@k from 2,000 source queries using 200,000 target sentences. We use the same embeddings as in (Smith et al., 2017). Their results are marked with the symbol †.

Refinement: closing the gap with supervised approaches The refinement step on the synthetic bilingual vocabulary constructed after adversarial training brings an additional and significant

gain in performance, closing the gap between our approach and the supervised baseline. In Table 4.1, we observe that our unsupervised method even outperforms our strong supervised baseline on en-it and en-es, and is able to retrieve the correct translation of a source word with up to 83% accuracy. The better performance of the unsupervised approach can be explained by the strong similarity of co-occurrence statistics between the languages, and by the limitation in the supervised approach that uses a pre-defined fixed-size vocabulary (of 5,000 unique source words): in our case the refinement step can potentially use more anchor points. In Table 4.3, we also observe a strong gain in accuracy (up to 15%) on sentence retrieval using bag-of-words embeddings, which is consistent with the gain observed on the word retrieval task.

Application to a low-resource language pair and to machine translation Our method is particularly suited for low-resource languages for which there only exists a very limited amount of parallel data. We apply it to the English-Esperanto language pair. We use the fastText embeddings trained on Wikipedia, and create a dictionary based on an online lexicon. The performance of our unsupervised approach on English-Esperanto is of 28.2%, compared to 29.3% with the supervised method. On Esperanto-English, our unsupervised approach obtains 25.6%, which is 1.3% better than the supervised method. The dictionary we use for that language pair does not take into account the polysemy of words, which explains why the results are lower than on other language pairs. People commonly report the P@5 to alleviate this issue. In particular, the P@5 for English-Esperanto and Esperanto-English is of 46.5% and 43.9% respectively.

SemEval 2017	en-es	en-de	en-it
<i>Methods with cross-lingual supervision</i>			
NASARI	0.64	0.60	0.65
our baseline	0.72	0.72	0.71
<i>Methods without cross-lingual supervision</i>			
Adv	0.69	0.70	0.67
Adv - Refine	0.71	0.71	0.71

Table 4.4: **Cross-lingual wordsim task.** NASARI (Camacho-Collados et al., 2016) refers to the official SemEval2017 baseline. We report Pearson correlation.

	en-eo	eo-en
Dictionary - NN	6.1	11.9
Dictionary - CSLS	11.1	14.3

Table 4.5: **BLEU score on English-Esperanto.** Although being a naive approach, word-by-word translation is enough to get a rough idea of the input sentence. The quality of the generated dictionary has a significant impact on the BLEU score.

To show the impact of such a dictionary on machine translation, we apply it to the English-Esperanto Tatoeba corpora (Tiedemann, 2012). We remove all pairs containing sentences with

unknown words, resulting in about 60k pairs. Then, we translate sentences in both directions by doing word-by-word translation. In Table 4.5, we report the BLEU score with this method, when using a dictionary generated using nearest neighbors, and CSLS. With CSLS, this naive approach obtains 11.1 and 14.3 BLEU on English-Esperanto and Esperanto-English respectively. Table 4.6 in the complementary analysis (see subsection below) shows some examples of sentences in Esperanto translated into English using word-by-word translation. As one can see, the meaning is mostly conveyed in the translated sentences, but the translations contain some simple errors. For instance, the “mi” is translated into “sorry” instead of “i”, etc. The translations could easily be improved using a language model.

4.1.5 Complementary analysis

In order to gain a better understanding of the impact of using similar corpora or similar word embedding methods, we investigated merging two English monolingual embedding spaces using either Wikipedia or the Gigaword corpus ((Parker et al., 2011)), and either Skip-Gram, CBOW or fastText methods (see Figure 4.3).

Source	mi kelkfoje parolas kun mia najbaro tra la barilo .
Hypothesis	sorry sometimes speaks with my neighbor across the barrier .
Reference	i sometimes talk to my neighbor across the fence .
Source	la viro malantaŭ ili ludas la pianon .
Hypothesis	the man behind they plays the piano .
Reference	the man behind them is playing the piano .
Source	bonvole protektu min kontraŭ tiuj malbonaj viroj .
Hypothesis	gratefully protects hi against those worst men .
Reference	please defend me from such bad men .

Table 4.6: **Esperanto-English.** Examples of fully unsupervised word-by-word translations. The translations reflect the meaning of the source sentences, and could potentially be improved using a simple language model.

4.1.6 Related work

In this section we make an in-depth review of the specific area of unsupervised word translation and word embeddings alignment that the related work in Section 2. Work on *bilingual lexicon induction* without parallel corpora has a long tradition, starting with the seminal works by Rapp (1995) and

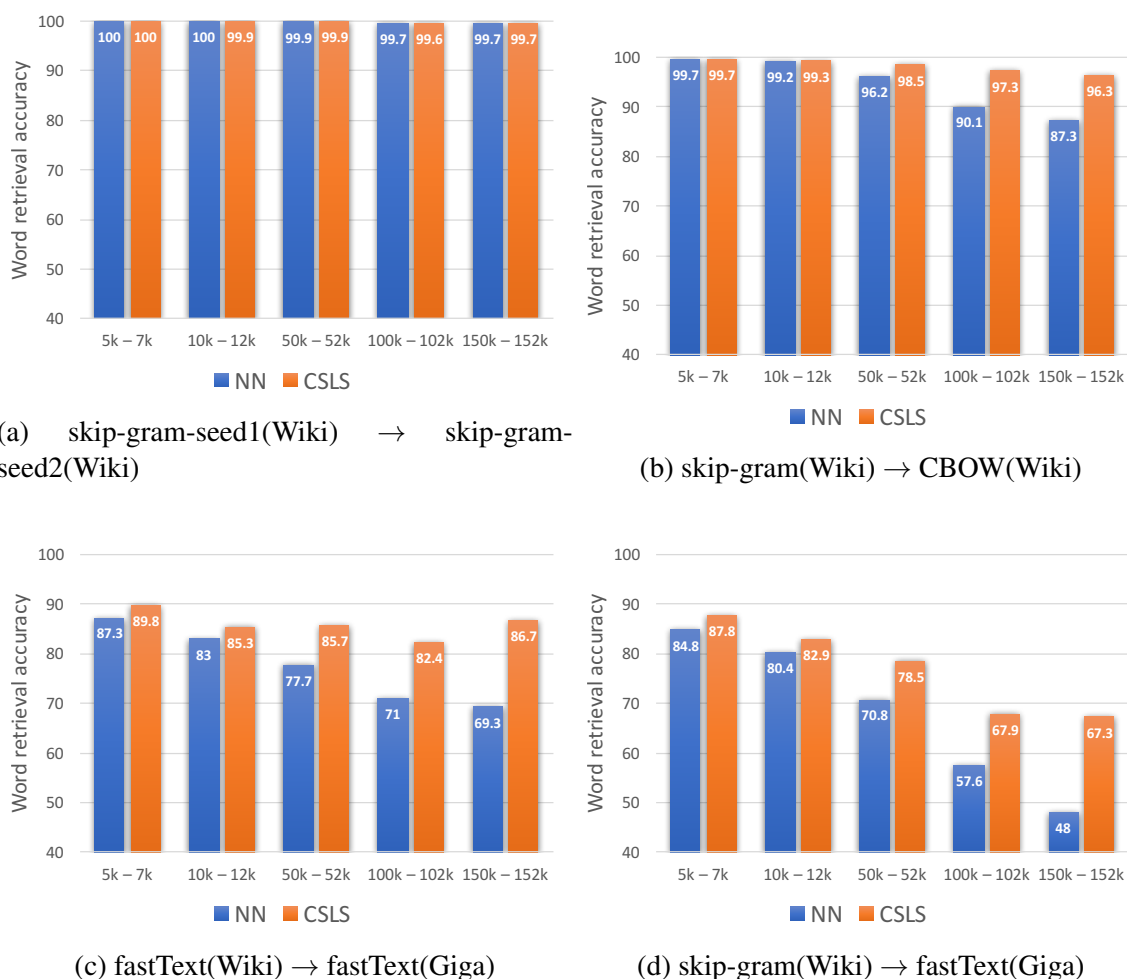


Figure 4.3: **English to English word alignment accuracy.** Evolution of word translation retrieval accuracy with regard to word frequency, using either Wikipedia (Wiki) or the Gigaword corpus (Giga), and either skip-gram, continuous bag-of-words (CBOW) or fastText embeddings. The model can learn to perfectly align embeddings trained on the same corpus but with different seeds (a), as well as embeddings learned using different models (overall, when employing CSLS which is more accurate on rare words) (b). However, the model has more trouble aligning embeddings trained on different corpora (Wikipedia and Gigaword) (c). This can be explained by the difference in co-occurrence statistics of the two corpora, particularly on the rarer words. Performance can be further deteriorated by using both different models and different types of corpus (d).

Fung (1995). Similar to our approach, they exploit the Harris (1954) distributional structure, but using discrete word representations such as TF-IDF vectors. Following studies by Fung and Yee

(1998); Rapp (1999); Schafer and Yarowsky (2002); Koehn and Knight (2002); Haghighi et al. (2008); Irvine and Callison-Burch (2013) leverage statistical similarities between two languages to learn small dictionaries of a few hundred words. These methods need to be initialized with a seed bilingual lexicon, using for instance the edit distance between source and target words. This can be seen as prior knowledge, only available for closely related languages. There is also a large amount of studies in statistical decipherment, where the machine translation problem is reduced to a deciphering problem, and the source language is considered as a ciphertext (Ravi and Knight, 2011; Pourdamghani and Knight, 2017). Although initially not based on distributional semantics, recent studies show that the use of word embeddings can bring significant improvement in statistical decipherment (Dou et al., 2015).

The rise of distributed word embeddings has revived some of these approaches, now with the goal of aligning embedding spaces instead of just aligning vocabularies. Cross-lingual word embeddings can be used to extract bilingual lexicons by computing the nearest neighbor of a source word, but also allow other applications such as sentence retrieval or cross-lingual document classification (Klementiev et al., 2012). In general, they are used as building blocks for various cross-lingual language processing systems. More recently, several approaches have been proposed to learn bilingual dictionaries mapping from the source to the target space (Mikolov et al., 2013b; Zou et al., 2013; Faruqui and Dyer, 2014; Ammar et al., 2016). In particular, Xing et al. (2015) showed that adding an orthogonality constraint to the mapping can significantly improve performance, and has a closed-form solution. This approach was further referred to as the Procrustes approach in Smith et al. (2017).

The hubness problem for cross-lingual word embedding spaces was investigated by Dinu et al. (2015). The authors added a correction to the word retrieval algorithm by incorporating a nearest neighbors reciprocity term. More similar to our cross-domain similarity local scaling approach, Smith et al. (2017) introduced the inverted-softmax to down-weight similarities involving often-retrieved hub words. Intuitively, given a query source word and a candidate target word, they estimate the probability that the candidate translates back to the query, rather than the probability that the query translates to the candidate.

Recent work by Smith et al. (2017) leveraged identical character strings in both source and target languages to create a dictionary with low supervision, on which they applied the Procrustes algorithm. Similar to this approach, recent work by Artetxe et al. (2017) used identical digits and numbers to form an initial seed dictionary, and performed an update similar to our refinement step, but iteratively until convergence. While they showed they could obtain good results using as little

as twenty parallel words, their method still needs cross-lingual information and is not suitable for languages that do not share a common alphabet. For instance, the method of Artetxe et al. (2017) on our dataset does not work on the word translation task for any of the language pairs, because the digits were filtered out from the datasets used to train the fastText embeddings. This iterative EM-based algorithm initialized with a seed lexicon has also been explored in other studies (Haghighi et al., 2008; Kondrak et al., 2017).

There has been a few attempts to align monolingual word vector spaces with no supervision. Similar to our work, Zhang et al. (2017a) employed adversarial training, but their approach is different than ours in multiple ways. First, they rely on sharp drops of the discriminator accuracy for model selection. In our experiments, their model selection criterion does not correlate with the overall model performance, as shown in Figure 4.2. Furthermore, it does not allow for hyper-parameters tuning, since it selects the best model over a single experiment. We argue it is a serious limitation, since the best hyper-parameters vary significantly across language pairs. Despite considering small vocabularies of a few thousand words, their method obtained weak results compared to supervised approaches. More recently, Zhang et al. (2017b) proposed to minimize the earth-mover distance after adversarial training. They compare their results only to their supervised baseline trained with a small seed lexicon, which is one to two orders of magnitude smaller than what we report here.

4.1.7 Conclusion

In this section, we show for the first time that one can align word embedding spaces without any cross-lingual supervision, solely based on unaligned datasets of each language, while reaching or outperforming the quality of previous supervised approaches in several cases. Using adversarial training, we are able to initialize a linear mapping between a source and a target space, which we also use to produce a synthetic parallel dictionary. It is then possible to apply the same techniques proposed for supervised techniques, namely a Procrustean optimization. Two key ingredients contribute to the success of our approach: First we propose a simple criterion that is used as an effective unsupervised validation metric. Second we propose the similarity measure CSLS, which mitigates the hubness problem and drastically increases the word translation accuracy. As a result, our approach produces high-quality dictionaries between different pairs of languages, with up to 83.3% precision@1 on the Spanish-English word translation task. This performance is on par with supervised approaches. Our method is also effective on the English-Esperanto pair, thereby showing that it works for low-resource language pairs, and can be used as a first step towards unsupervised

machine translation. Indeed, we used our unsupervised word embeddings alignment as an initialization step for fully unsupervised machine translation (at the sentence level). In Lample et al. (2016, 2018b) we use a neural and a phrase-based machine translation approach that only requires monolingual corpora, and we show that the initialization of the lookup table of the LSTM/Transformer encoder and decoder by the unsupervised cross-lingual word embeddings obtained through MUSE is essential. We will come back to unsupervised machine translation in Section 4.3, where we show that we can improve even more the pretraining of encoders and decoders for unsupervised machine translation. We will see that our approach leads to a score of 33 BLEU on English-French without using any parallel data, a score that is on par with those obtained by the best supervised methods near 2014.

4.2 Evaluating Cross-Lingual Sentence Representations

In this section, we move from the word level to the sentence level and build the first large-scale cross-lingual sentence classification benchmark which helps us evaluate our cross-lingual sentence encoders. State-of-the-art natural language processing systems often rely on supervision in the form of annotated data to learn competent models. These models are generally trained on data in a single language (usually English), and cannot be directly used beyond that language. Since collecting data in every language is not realistic, there has been a growing interest in cross-lingual language understanding (XLU) and low-resource cross-language transfer. In this section, we construct an evaluation set for XLU by extending the development and test sets of the Multi-Genre Natural Language Inference Corpus (MultiNLI) to 15 languages, including low-resource languages such as Swahili and Urdu. We hope that our dataset, dubbed XNLI, will catalyze research in cross-lingual sentence understanding by providing an informative standard evaluation task. In addition, we provide several baselines for multilingual sentence understanding, including two based on machine translation systems, and two that use parallel data to train aligned multilingual bag-of-words and LSTM encoders. We find that XNLI represents a practical and challenging evaluation suite, and that directly translating the test data yields the best performance among available baselines.

4.2.1 Introduction

Contemporary natural language processing systems typically rely on annotated data to learn how to perform a task (e.g., classification, sequence tagging, natural language inference). Most commonly the available training data is in a single language (e.g., English or Chinese) and the resulting system can perform the task only in the training language. In practice, however, systems used in major international products need to handle inputs in many languages. In these settings, it is nearly impossible to annotate data in all languages that a system might encounter during operation.

A scalable way to build multilingual systems is through cross-lingual language understanding (XLU), in which a system is trained primarily on data in one language and evaluated on data in others. While XLU shows promising results for tasks such as cross-lingual document classification (Klementiev et al., 2012; Schwenk and Li, 2018), there are very few, if any, XLU benchmarks for more difficult language understanding tasks like natural language inference. NLI has emerged as a practical test bed for work on sentence understanding. In NLI, a system is tasked with reading two sentences and determining whether one entails the other, contradicts it, or neither (*neutral*). Recent crowdsourced annotation efforts have yielded datasets for NLI in English (Bowman et al., 2015;

Language	Premise / Hypothesis	Genre	Label
English	You don't have to stay there. You can leave.	Face-To-Face	Entailment
French	La figure 4 montre la courbe d'offre des services de partage de travaux. Les services de partage de travaux ont une offre variable.	Government	Entailment
Spanish	Y se estremeció con el recuerdo. El pensamiento sobre el acontecimiento hizo su estremecimiento.	Fiction	Entailment
German	Während der Depression war es die ärmste Gegend, kurz vor dem Hungertod. Die Weltwirtschaftskrise dauerte mehr als zehn Jahre an.	Travel	Neutral
Swahili	Ni silaha ya plastiki ya moja kwa moja inayopiga risasi. Inadumu zaidi kuliko silaha ya chuma.	Telephone	Neutral
Russian	И мы занимаемся этим уже на протяжении 85 лет. Мы только начали этим заниматься.	Letters	Contradiction
Chinese	让我告诉你，美国人最终如何看待你作为独立顾问的表现。 美国人完全不知道您是独立律师。	Slate	Contradiction
Arabic	تحتاج الوكالات لأن تكون قادرة على قياس مستويات النجاح. لا يمكننا للوكالات أن التعرف ما إذا كانت ناجحة أم لا	Nine-Eleven	Contradiction

Table 4.7: Examples (premise and hypothesis) from various languages and genres from the XNLI corpus.

Williams et al., 2018) with nearly a million examples, and these have been widely used to evaluate neural network architectures and training strategies (Rocktäschel et al., 2016; Gong et al., 2018; Peters et al., 2018; Wang et al., 2018), as well as to train effective, reusable sentence representations (Conneau et al., 2017; Subramanian et al., 2018; Cer et al., 2018; Conneau et al., 2018a).

In this section, we introduce a benchmark that we call the Cross-lingual Natural Language Inference corpus, or XNLI, by extending these NLI corpora to 15 languages. XNLI consists of 7500 human-annotated development and test examples in NLI three-way classification format in English, French, Spanish, German, Greek, Bulgarian, Russian, Turkish, Arabic, Vietnamese, Thai, Chinese, Hindi, Swahili and Urdu, making a total of 112,500 annotated pairs. These languages span several language families, and with the inclusion of Swahili and Urdu, include two lower-resource languages as well. Because of its focus on development and test data, this corpus is designed to evaluate *cross-lingual* sentence understanding, where models have to be trained in one language and tested in different ones. We evaluate several approaches to cross-lingual learning of natural language inference that leverage parallel data from publicly available corpora at training time. We show that parallel data can help align sentence encoders in multiple languages such that a classifier trained with English NLI data can correctly classify pairs of sentences in other languages. While

outperformed by our machine translation baselines, we show that this alignment mechanism gives very competitive results. A second practical use of XNLI is the evaluation of pretrained general-purpose language-universal sentence encoders. We hope that this benchmark will help the research community build multilingual text embedding spaces. Such embeddings spaces will facilitate the creation of multilingual systems that can transfer across languages with little or no extra supervision.

The chapter is organized as follows: We next survey the related literature on cross-lingual language understanding. We then describe our data collection methods and the resulting corpus in Section 4.2.3. We describe our baselines in Section 4.2.4, and finally present and discuss results in Section 4.2.5.

4.2.2 Related Work

In this section, we review the related work specific to cross-lingual understanding and cross-lingual text representations in particular.

Multilingual Word Embeddings Much of the work on multilinguality in language understanding has been at the word level. Several approaches have been proposed to learn cross-lingual word representations, i.e., word representations where translations are close in the embedding space. Many of these methods require some form of supervision (typically in the form of a small bilingual lexicon) to align two sets of source and target embeddings to the same space (Mikolov et al., 2013b; Kociský et al., 2014; Faruqui and Dyer, 2014; Ammar et al., 2016). More recent studies have showed that cross-lingual word embeddings can be generated with no supervision whatsoever (Artetxe et al., 2017; Conneau et al., 2018b) as explained in the previous section.

Sentence Representation Learning Many approaches have been proposed to extend word embeddings to sentence or paragraph representations (Le and Mikolov, 2014; Wieting et al., 2015; Arora et al., 2017). The most straightforward way to generate sentence embeddings is to consider an average or weighted average of word representations, usually referred to as continuous bag-of-words (CBOW). Although naïve, this method often provides a strong baseline. More sophisticated approaches—such as the unsupervised SkipThought model of Kiros et al. (2015) that extends the skip-gram model of Mikolov et al. (2013c) to the sentence level—have been proposed to capture syntactic and semantic dependencies inside sentence representations. While these fixed-size sentence embedding methods have been outperformed by their supervised counterparts (Conneau et al., 2017; Subramanian et al., 2018) as shown in Section 3, some recent developments have shown that

pretrained language models can also transfer very well, either when the hidden states of the model are used as contextualized word vectors (Peters et al., 2018), or when the full model is fine-tuned on transfer tasks (Radford et al., 2018; Howard and Ruder, 2018).

Multilingual Sentence Representations In the past sections, we have seen ways to build strong monolingual sentence representations, and how to align word embedding spaces. Our goal in this section is to extend those ideas to build strong cross-lingual representations of sentences. There has been some effort on developing multilingual sentence embeddings. For example, Chandar et al. (2013) train bilingual autoencoders with the objective of minimizing reconstruction error between two languages. Schwenk et al. (2017) and España-Bonet et al. (2017) jointly train a sequence-to-sequence MT system on multiple languages to learn a shared multilingual sentence embedding space. Hermann and Blunsom (2014) propose a compositional vector model involving unigrams and bigrams to learn document level representations. Pham et al. (2015a) directly train embedding representations for sentences with no attempt at compositionality. Zhou et al. (2016c) learn bilingual document representations by minimizing the Euclidean distance between document representations and their translations.

Cross-lingual Evaluation Benchmarks The lack of evaluation benchmark has hindered the development of such multilingual representations. Most previous approaches use the Reuters cross-lingual document classification corpus Klementiev et al. (2012) for evaluation. However, the classification in this corpus is done at document level, and, as there are many ways to aggregate sentence embeddings, the comparison between different sentence embeddings is difficult. Moreover, the distribution of classes in the Reuters corpus is highly unbalanced, and the dataset does not provide a development set in the target language, further complicating experimental comparisons.

In addition to the Reuters corpus, Cer et al. (2017) propose sentence-level multilingual training and evaluation datasets for semantic textual similarity in four languages. There have also been efforts to build multilingual RTE datasets, either through translating English data (Mehdad et al., 2011), or annotating sentences from a parallel corpora (Negri et al., 2011). More recently, Agić and Schlueter (2018) provide a corpus, that is very complementary to our work, of human translations for 1332 pairs of the SNLI data into Arabic, French, Russian, and Spanish. Among all these benchmarks, XNLI is the first large-scale corpus for evaluating sentence-level representations on that many languages.

In practice, cross-lingual sentence understanding goes beyond translation. For instance, Mohammad et al. (2016) analyze the differences in human sentiment annotations of Arabic sentences and their English translations, and conclude that most of them come from cultural differences. Similarly, Smith et al. (2016) show that most of the degradation in performance when applying a classification model trained in English to Spanish data translated to English is due to cultural differences. One of the limitations of the XNLI corpus is that it does not capture these differences, since it was obtained by translation. We see the XNLI evaluation as a necessary step for multilingual NLP before tackling the even more complex problem of domain-adaptation that occurs when handling this change in style from one language to another.

4.2.3 The XNLI Corpus

Because the test portion of the Multi-Genre NLI data was kept private, the Cross-lingual NLI Corpus (XNLI) is based on new English NLI data. To collect the core English portion, we follow precisely the same crowdsourcing-based procedure used for the existing Multi-Genre NLI corpus, and collect and validate 750 new examples from each of the ten text sources used in that corpus for a total of 7500 examples. With that portion in place, we create the full XNLI corpus by employing professional translators to translate it into our fourteen target languages. This section describes this process and the resulting corpus.

Translating, rather than generating new hypothesis sentences in each language separately, has multiple advantages. First, it ensures that the data distributions are maximally similar across languages. As speakers of different languages may have slightly different intuitions about how to fill in the supplied prompt, this allows us to avoid adding this unwanted degree of freedom. Second, it allows us to use the same trusted pool of workers as was used in prior NLI crowdsourcing efforts, without the need for training a new pool of workers in each language. Third, for any premise, this process allows us to have a corresponding hypothesis in any language. XNLI can thus potentially be used to evaluate whether an Arabic or Urdu premise is entailed with a Bulgarian or French hypothesis etc. This results in more than 1.5M combinations of hypothesis and premises. Note that we do not consider that use case in this work.

This translation approach carries with it the risk that the semantic relations between the two sentences in each pair might not be reliably preserved in translation, as Mohammad et al. (2016) observed for sentiment. We investigate this potential issue in our corpus and find that, while it does occur, it only concerns a negligible number of sentences (see below).

Data Collection

The English Corpus Our collection procedure for the English portion of the XNLI corpus follows the same procedure as the MultiNLI corpus. We sample 250 sentences from each of the ten sources that were used in that corpus, ensuring that none of those selected sentences overlap with the distributed corpus. Nine of the ten text sources are drawn from the second release of the Open American National Corpus⁴: *Face-To-Face*, *Telephone*, *Government*, *9/11*, *Letters*, *Oxford University Press (OUP)*, *Slate*, *Verbatim*, and *Government*. The tenth, *Fiction*, is drawn from the novel *Captain Blood* (Sabatini, 1922). We refer the reader to Williams et al. (2018) for more details on each genre.

Given these sentences, we ask the same MultiNLI worker pool from a crowdsourcing platform to produce three hypotheses for each premise, one for each possible label.

We present premise sentences to workers using the same templates as were used in MultiNLI. We also follow that work in pursuing a second validation phase of data collection in which each pair of sentences is relabeled by four other workers. For each validated sentence pair, we assign a gold label representing a majority vote between the initial label assigned to the pair by the original annotator, and the four additional labels assigned by validation annotators. We obtained a three-vote consensus for 93% of the data. In our experiments, we kept the 7% additional ones, but we mark these ones with a special label '-'.⁴

	en	fr	es	de	el	bg	ru	tr	ar	vi	th	zh	hi	sw	ur
Premise	21.7	24.1	22.1	21.1	21.0	20.9	19.6	16.8	20.7	27.6	22.1	21.8	23.2	18.7	24.1
Hypothesis	10.7	12.4	10.9	10.8	10.6	10.4	9.7	8.4	10.2	13.5	10.4	10.8	11.9	9.0	12.3

Table 4.8: Average number of tokens per sentence in the XNLI corpus for each language.

Translating the Corpus Finally, we hire translators to translate the resulting sentences into 15 languages using the *One Hour Translation* platform. We translate the premises and hypotheses separately, to ensure that no context is added to the hypothesis that was not there originally, and simply copy the labels from the English source text. Some development examples are shown in Table 4.7.

⁴<http://www.anc.org/>

The Resulting Corpus

One main concern in studying the resulting corpus is to determine whether the gold label for some of the sentence pairs changes as a result of information added or removed in the translation process. Investigating the data manually, we find an example in the Chinese translation where an entailment relation becomes a contradictory relation, while the entailment is preserved in other languages. Specifically, the term *upright* which was used in English as entailment of *standing*, was translated into Chinese as *sitting upright* thus creating a contradiction. However, the difficulty of finding such an example in the data suggests its rarity.

To quantify this observation, we recruit two bilingual annotators to re-annotate 100 examples each in both English and French following our standard validation procedure. The examples are drawn from two non-overlapping random subsets of the development data to prevent the annotators from seeing the source English text for any translated text they annotate. With no training or burn-in period, these annotators recover the English consensus label 85% of the time on the original English data and 83% of the time on the translated French, suggesting that the overall semantic relationship between the two languages has been preserved. As most sentences are relatively easy to translate, in particular the hypotheses generated by the workers, there seems to be little ambiguity added by the translator.

More broadly, we find that the resulting corpus has similar properties to the MultiNLI corpus. For all languages, on average, the premises are twice as long as the hypotheses (See Table 4.8). The top hypothesis words indicative of the class label – scored using the mutual information between each word and class in the corpus – are similar across languages, and overlap those of the MultiNLI corpus (Gururangan et al., 2018). For example, a translation of at least one of the words *no*, *not* or *never* is among the top two cues for contradiction in all languages.

As in the original MultiNLI corpus, we expect that cues like these (‘artifacts’, in Gururangan’s terms, also observed by Poliak et al., 2018; Tsuchiya, 2018) allow a baseline system to achieve better-than-random accuracy with access only to the premise sentences. We accept this as an unavoidable property of the NLI task over naturalistic sentence pairs, and see no reason to expect that this baseline would achieve better accuracy than the relatively poor 53% seen in Gururangan et al. (2018).

The current version of the corpus is freely available⁵⁶ for typical machine learning uses, and may be modified and redistributed. The majority of the corpus sentences are released under the

⁵<http://nyu.edu/projects/bowman/xnli/>

⁶<http://github/facebookresearch/xnli>

OANC’s license which allows all content to be freely used, modified, and shared under permissive terms. The data in the *Fiction* genre from *Captain Blood* are in the public domain in the United States (but may be licensed differently elsewhere).

4.2.4 Cross-Lingual NLI

In this section we present results with XLU systems that can serve as baselines, in particular for section 4.3.

Translation-Based Approaches

The most straightforward techniques for XLU rely on translation systems. There are two natural ways to use a translation system: TRANSLATE TRAIN, where the training data is translated into each target language to provide data to train each classifier, and TRANSLATE TEST, where a translation system is used at test time to translate input sentences to the training language. These two methods provide strong baselines, but both present practical challenges. The former requires training and maintaining as many classifiers as there are languages, while the latter relies on computationally-intensive translation at test time. Both approaches are limited by the quality of the translation system, which itself varies with the quantity of available training data and the similarity of the language pair involved.

Multilingual Sentence Encoders

An alternative to translation is to rely on language-universal embeddings of text and build multilingual classifiers on top of these representations. If an encoder produces an embedding of an English sentence close to the embedding of its translation in another language, then a classifier learned on top of English sentence embeddings will be able to classify sentences from different languages without needing a translation system at inference time.

We evaluate two types of cross-lingual sentence encoders: (i) pretrained universal multilingual sentence embeddings based on the average of word embeddings (X-CBOW), (ii) bidirectional-LSTM (Hochreiter and Schmidhuber, 1997) sentence encoders trained on the MultiNLI training data (X-BILSTM). The former evaluates transfer learning while the latter evaluates NLI-specific encoders trained on in-domain data. Both approaches use the same alignment loss for aligning sentence embedding spaces from multiple languages which is present below. We consider two

ways of extracting feature vectors from the BiLSTM: either using the initial and final hidden states (Sutskever et al., 2014), or using the element-wise max over all states (Collobert and Weston, 2008).

The first approach is commonly used as a strong baseline for monolingual sentence embeddings (Arora et al., 2017; Conneau and Kiela, 2018; Gouews et al., 2014). Concretely, we consider the English fastText word embedding space as being fixed, and fine-tune embeddings in other languages so that the average of the word vectors in a sentence is close to the average of the word vectors in its English translation. The second approach consists in learning an English sentence encoder on the MultiNLI training data along with an encoder on the target language, with the objective that the representations of two translations are nearby in the embedding space. In both approaches, an English encoder is fixed, and we train target language encoders to match the output of this encoder. This allows us to build sentence representations that belong to the same space. Joint training of encoders and parameter sharing are also interesting directions to improve and simplify the alignment of sentence embedding spaces. We explore this research direction in section 4.3.

In all experiments, we consider encoders that output a vector of fixed size as a sentence representation. While previous work shows that performance on the NLI task can be improved by using cross-sentence attention between the premise and hypothesis (Rocktäschel et al., 2016; Gong et al., 2018), we focus on methods with fixed-size sentence embeddings.

Aligning Word Embeddings Multilingual word embeddings are an efficient way to transfer knowledge from one language to another. For instance, Zhang et al. (2016) show that cross-lingual embeddings can be used to extend an English part-of-speech tagger to the cross-lingual setting, and Xiao and Guo (2014) achieve similar results in dependency parsing. Cross-lingual embeddings also provide an efficient mechanism to bootstrap neural machine translation (NMT) systems for low-resource language pairs, which is critical in the case of unsupervised machine translation (Lample et al., 2018a; Artetxe et al., 2018; Lample et al., 2018b). In that case, the use cross-lingual embeddings directly helps the alignment of sentence-level encoders. In this section, we pretrain our embeddings using the common-crawl word embeddings (Grave et al., 2018) aligned with the MUSE library of Conneau et al. (2018b).

Universal Multilingual Sentence Embeddings Most of the successful recent approaches for learning universal sentence representations have relied on English (Kiros et al., 2015; Arora et al., 2017; Conneau et al., 2017; Subramanian et al., 2018; Cer et al., 2018) as we have seen in section 3. While notable recent approaches have considered building a shared sentence encoder for multiple

languages using publicly available parallel corpora (Johnson et al., 2017b; Schwenk et al., 2017; España-Bonet et al., 2017), the lack of a large-scale, sentence-level semantic evaluation has limited their adoption by the community. In particular, these methods did not cover the scale of languages considered in XNLI, and are limited to high-resource languages. We will see in section 4.3 that new methods were created after our work on XNLI such as the one of Artetxe and Schwenk (2018) and Lample and Conneau (2019) that cover a wide range of languages. As a baseline for the evaluation of pretrained multilingual sentence representations in the 15 languages of XNLI, in this section we consider state-of-the-art common-crawl embeddings with a CBOW encoder. Our approach, dubbed X-CBOW, consists in fixing the English pretrained word embeddings, and fine-tuning the target (e.g., French) word embeddings so that the CBOW representations of two translations are close in embedding space. In that case, we consider our multilingual sentence embeddings as being pretrained and only learn a classifier on top of them to evaluate their quality, similar to the “transfer” tasks of SentEval (see section 3.1) but in the multilingual setting.

Aligning Sentence Embeddings Training for similarity of source and target sentences in an embedding space is conceptually and computationally simpler than generating a translation in the target language from a source sentence. We propose a method for training for cross-lingual similarity and evaluate approaches based on the simpler task of aligning sentence representations. Under our objective, the embeddings of two parallel sentences need not be identical, but only close enough in the embedding space that the decision boundary of the English classifier captures the similarity.

We propose a simple alignment loss function to align the embedding spaces of two different languages. Specifically, we train an English encoder on NLI, and train a target encoder by minimizing the loss:

$$\mathcal{L}_{\text{align}}(x, y) = \text{dist}(x, y) - \lambda(\text{dist}(x_c, y) + \text{dist}(x, y_c))$$

where (x, y) corresponds to the source and target sentence embeddings, (x_c, y_c) is a contrastive term (i.e. negative sampling), λ controls the weight of the negative examples in the loss. For the distance measure, we use the L2 norm $\text{dist}(x, y) = \|x - y\|_2$. A ranking loss (Weston et al., 2011) of the form

$$\begin{aligned} \mathcal{L}_{\text{rank}}(x, y) = & \max(0, \alpha - \text{dist}(x, y_c) + \text{dist}(x, y)) + \\ & \max(0, \alpha - \text{dist}(x_c, y) + \text{dist}(x, y)) \end{aligned}$$

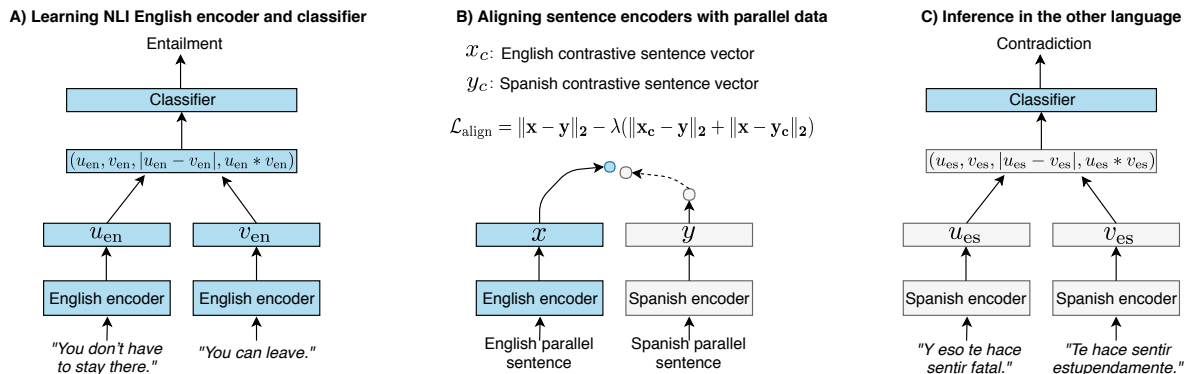


Figure 4.4: **Illustration of language adaptation by sentence embedding alignment.** A) The English encoder and classifier in blue are learned on English (*in-domain*) NLI data. The encoder can also be pretrained (*transfer learning*). B) The Spanish encoder in gray is trained to mimic the English encoder using parallel data. C) After alignment of the encoders, the classifier can make predictions for Spanish.

that pushes the sentence embeddings of a translation pair to be closer than the ones of negative pairs leads to very poor results in this particular case. As opposed to \mathcal{L}_{align} , \mathcal{L}_{rank} does not force the embeddings of sentence pairs to be close enough so that the shared classifier can understand that these sentences have the same meaning.

We use \mathcal{L}_{align} in the cross-lingual embeddings baselines X-CBOW, X-BILSTM-LAST and X-BILSTM-MAX. For X-CBOW, the encoder is pretrained and not fine-tuned on NLI (transfer-learning), while the English X-BiLSTMs are trained on the MultiNLI training set (in-domain). For the three methods, the English encoder and classifier are then fixed. Each of the 14 other languages have their own encoders with same architecture. These encoders are trained to "copy" the English encoder using the \mathcal{L}_{align} loss and the parallel data described in section 4.2.5. Our sentence embedding alignment approach is illustrated in Figure 4.4.

We only back-propagate through the target encoder when optimizing \mathcal{L}_{align} such that all 14 encoders live in the same English embedding space. In these experiments, we initialize lookup tables of the LSTMs with pretrained cross-lingual embeddings discussed in Section 4.1.

	fr	es	de	el	bg	ru	tr	ar	vi	th	zh	hi	sw	ur
XX-En BLEU	41.2	45.8	39.3	42.1	38.7	27.1	29.9	35.2	23.6	22.6	24.6	27.3	21.3	24.4
En-XX BLEU	49.3	48.5	38.8	42.4	34.2	24.9	21.9	15.8	39.9	21.4	23.2	37.5	24.6	24.1
Word translation P@1	73.7	73.9	65.9	61.1	61.9	60.6	55.0	51.9	35.8	25.4	48.6	48.2	-	-

Table 4.9: BLEU scores of our translation models (XX-En) P@1 for multilingual word embeddings.

4.2.5 Experiments and Results

Training details

We use internal translation systems to translate data between English and the 10 other languages. For TRANSLATE TEST (see Table 4.10), we translate each test set into English, while for the TRANSLATE TRAIN, we translate the English training data of MultiNLI.⁷ To give an idea of the translation quality, we give BLEU scores of the automatic translation from the foreign language into English of the XNLI test set in Table 4.9. We use the MOSES tokenizer for most languages, falling back on the default English tokenizer when necessary. We use the Stanford segmenter for Chinese (Chang et al., 2008), and the *pythainlp* package for Thai.

We use pretrained 300D aligned word embeddings for both X-CBOW and X-BILSTM and only consider the most 500,000 frequent words in the dictionary, which generally covers more than 98% of the words found in XNLI corpora. We set the number of hidden units of the BiLSTMs to 512, and use the Adam optimizer (Kingma and Ba, 2014) with default parameters. As for InferSent in section 3.2, the classifier receives a vector $[u, v, |u - v|, u * v]$, where u and v are the embeddings of the premise and hypothesis provided by the shared encoder, and $*$ corresponds to the element-wise multiplication (see Figure 4.4). For the alignment loss, setting λ to 0.25 worked best in our experiments, and we found that the trade-off between the importance of the positive and the negative pairs was particularly important (see Table 4.11). We sample negatives randomly. When fitting the target BiLSTM encoder to the English encoder, we fine-tune the lookup table associated to the target encoder, but keep the source word embeddings fixed. The classifier is a feed-forward neural network with one hidden layer of 128 hidden units, regularized with dropout (Srivastava et al., 2014) at a rate of 0.1. For X-BiLSTMs, we perform model selection on the XNLI validation set in each target language. For X-CBOW, we keep a validation set of parallel sentences to evaluate our alignment loss. The alignment loss requires a parallel dataset of sentences for each pair of languages, which

⁷To allow replication of results, we share the MT translations of XNLI training and test sets.

	en	fr	es	de	el	bg	ru	tr	ar	vi	th	zh	hi	sw	ur
<i>Machine translation baselines (TRANSLATE TRAIN)</i>															
BiLSTM-last	71.0	66.7	67.0	65.7	65.3	65.6	65.1	61.9	63.9	63.1	61.3	65.7	61.3	55.2	55.2
BiLSTM-max	73.7	68.3	68.8	66.5	66.4	67.4	66.5	64.5	65.8	66.0	62.8	67.0	62.1	58.2	56.6
<i>Machine translation baselines (TRANSLATE TEST)</i>															
BiLSTM-last	71.0	68.3	68.7	66.9	67.3	68.1	66.2	64.9	65.8	64.3	63.2	66.5	61.8	60.1	58.1
BiLSTM-max	73.7	70.4	70.7	68.7	69.1	70.4	67.8	66.3	66.8	66.5	64.4	68.3	64.2	61.8	59.3
<i>Evaluation of XNLI multilingual sentence encoders (in-domain)</i>															
X-BiLSTM-last	71.0	65.2	67.8	66.6	66.3	65.7	63.7	64.2	62.7	65.6	62.7	63.7	62.8	54.1	56.4
X-BiLSTM-max	73.7	67.7	68.7	67.7	68.9	67.9	65.4	64.2	64.8	66.4	64.1	65.8	64.1	55.7	58.4
<i>Evaluation of pretrained multilingual sentence encoders (transfer learning)</i>															
X-CBOW	64.5	60.3	60.7	61.0	60.5	60.4	57.8	58.7	57.5	58.8	56.9	58.8	56.3	50.4	52.2

Table 4.10: XNLI test accuracy for the 15 languages.

we describe next.

Parallel Datasets

We use publicly available parallel datasets to learn the alignment between English and target encoders. For French, Spanish, Russian, Arabic and Chinese, we use the United Nation corpora (Ziemski et al., 2016), for German, Greek and Bulgarian, the Europarl corpora (Koehn, 2005), for Turkish, Vietnamese and Thai, the OpenSubtitles 2018 corpus (Tiedemann, 2012), and for Hindi, the IIT Bombay corpus (Anoop et al., 2018). For all the above language pairs, we were able to gather more than 500,000 parallel sentences, and we set the maximum number of parallel sentences to 2 million. For the lower-resource languages Urdu and Swahili, the number of parallel sentences is an order of magnitude smaller than for the other languages we consider. For Urdu, we used the Bible and Quran transcriptions (Tiedemann, 2012), the OpenSubtitles 2016 (Pierre and Jörg, 2016) and 2018 corpora and LDC2010T21, LDC2010T23 LDC corpora, and obtained a total of 64k parallel sentences. For Swahili, we were only able to gather 42k sentences using the Global Voices corpus and the Tanzil Quran transcription corpus⁸.

Analysis

Comparing in-language performance in Table 4.10, we observe that, when using BiLSTMs, results are consistently better when we take the dimension-wise maximum over all hidden states

⁸<http://opus.nlpl.eu/>

(BiLSTM-max) compared to taking the last hidden state (BiLSTM-last), as observed in section 3.2 for InferSent. Unsurprisingly, BiLSTM results are better than the pretrained CBOW approach for all languages. As in Bowman et al. (2015), we also observe the superiority of BiLSTM encoders over CBOW, even when fine-tuning the word embeddings of the latter on the MultiNLI training set, thereby again confirming that the NLI task requires more than just word information. Both of these findings confirm our results obtained in section 3.2.

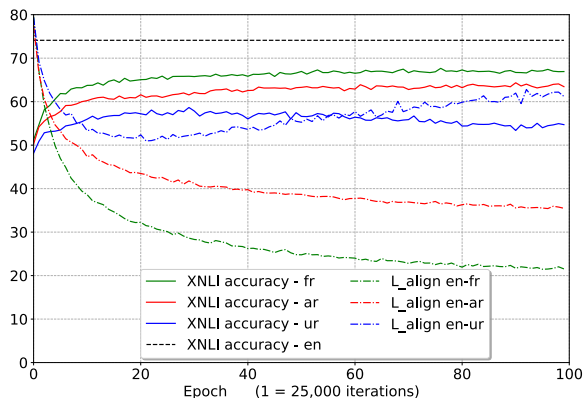


Figure 4.5: Evolution along training of alignment losses and X-BiLSTM XNLI French (fr), Arabic (ar) and Urdu (ur) accuracies. Observe the correlation between \mathcal{L}_{align} and accuracy.

Table 4.10 shows that translation offers a strong baseline for XLU. Within translation, TRANSLATE TEST appears to perform consistently better than TRANSLATE TRAIN for all languages. The best cross-lingual results in our evaluation are obtained by the TRANSLATE TEST approach for all cross-lingual directions. Within the translation approaches, as expected, we observe that cross-lingual performance depends on the quality of the translation system. In fact, translation-based results are very well-correlated with the BLEU scores for the translation systems; XNLI performance for three of the four languages with the best translation systems (comparing absolute BLEU, Table 4.9) is above 70%. This performance is still about three points below the English NLI performance of 73.7%. This slight drop in performance may be related to translation error, changes in style, or artifacts introduced by the machine translation systems that result in discrepancies between the training and test data.

For cross-lingual performance, we observe a healthy gap between the English results and the results obtained on other languages. For instance, for French, we obtain 67.7% accuracy when classifying French pairs using our English classifier and multilingual sentence encoder. When using our alignment process, our method is competitive with the TRANSLATE TRAIN baseline, suggesting

that it might be possible to encode similarity between languages directly in the embedding spaces generated by the encoders. However, these methods are still below the other machine translation baseline TRANSLATE TEST, which significantly outperforms the multilingual sentence encoder approach by up to 6% (Swahili). These production systems have been trained on much larger training data than the ones used for the alignment loss (section 4.2.5), which can partly explain the superiority of this method over the baseline. At inference time, the multilingual sentence encoder approach is however much cheaper than the TRANSLATE TEST baseline, and this method also does not require any machine translation system. Interestingly, the two points difference in accuracy between X-BiLSTM-last and X-BiLSTM-max is maintained across languages, which suggests that having a stronger encoder in English also positively impacts the transfer results on other languages.

For X-BiLSTM French, Urdu and Arabic encoders, we plot in Figure 4.5 the evolution of XNLI dev accuracies and the alignment losses during training. The latter are computed using XNLI parallel dev sentences. We observe a strong correlation between the alignment losses and XNLI accuracies. As the alignment on English-Arabic gets better for example, so does the accuracy on XNLI-ar. One way to understand this is to recall that the English classifier takes as input the vector $[u, v, |u - v|, u * v]$ where u and v are the embeddings of the premise and hypothesis. So this correlation between $\mathcal{L}_{\text{align}}$ and the accuracy suggests that, as English and Arabic embeddings $[u_{\text{en}}, v_{\text{en}}]$ and $[u_{\text{ar}}, v_{\text{ar}}]$ get closer for parallel sentences (in the sense of the L2-norm), the English classifier gets better at understanding Arabic embeddings $[u_{\text{ar}}, v_{\text{ar}}, |u_{\text{ar}} - v_{\text{ar}}|, u_{\text{ar}} * v_{\text{ar}}]$ and thus the accuracy improves. We observe some over-fitting for Urdu, which can be explained by the small number of parallel sentences (64k) available for that language.

In Table 4.11, we report the validation accuracy using BiLSTM-max on three languages with different training hyper-parameters. Fine-tuning the embeddings does not significantly impact the results, suggesting that the LSTM alone is ensuring alignment of parallel sentence embeddings. We also observe that the negative term is not critical to the performance of the model, but can lead to slight improvement in Chinese (up to 1.6%).

4.2.6 Conclusion

A typical problem in industrial applications is the lack of supervised data for languages other than English, and particularly for low-resource languages. Since annotating data in every language is not a realistic approach, there has been a growing interest in cross-lingual understanding and low-resource transfer in multilingual scenarios. In this work, we extend the development and test sets

	fr	ru	zh
$ft = 1, \lambda = 0.25$ [default]	68.9	66.4	67.9
$ft = 1, \lambda = 0.0$ (no negatives)	67.8	66.2	66.3
$ft = 1, \lambda = 0.5$	64.5	61.3	63.7
$ft = 0, \lambda = 0.25$	68.5	66.3	67.7

Table 4.11: **Validation accuracy using BiLSTM-max.** Default setting corresponds to $\lambda = 0.25$ (importance of the negative terms) and uses fine-tuning of the target lookup table ($ft = 1$).

of the Multi-Genre Natural Language Inference Corpus to 15 languages, including low-resource languages such as Swahili and Urdu. Our dataset, dubbed XNLI, is designed to address the lack of standardized evaluation protocols in cross-lingual understanding, and will hopefully help the community make further strides in this area. We present several approaches based on cross-lingual sentence encoders and machine translation systems. While machine translation baselines obtained the best results in our experiments, these approaches rely on computationally-intensive translation models either at training or at test time. We found that cross-lingual encoder baselines provide an encouraging and efficient alternative, and that further work was required to match the performance of translation based methods. After the publication of this work, XNLI was used in a series of work including Devlin et al. (2018); Artetxe and Schwenk (2018) and our own work on cross-lingual language model (Lample and Conneau, 2019). We discuss in more details in the next section these approaches that significantly outperform the results presented in this section.

4.3 Cross-lingual Language Models

Recent studies have demonstrated the efficiency of generative pretraining for English natural language understanding (Radford et al., 2018; Devlin et al., 2018). In this section, we extend this approach to multiple languages and show the effectiveness of cross-lingual pretraining. Similar in principle to the unsupervised alignment of word embeddings presented in section 4.1, we align sentence embedding spaces in a completely unsupervised way by leveraging cross-lingual pretraining. Specifically, we propose two methods to learn cross-lingual language models (XLMs): one unsupervised that only relies on monolingual data, and one supervised that leverages parallel data with a new cross-lingual language model objective. We obtain state-of-the-art results on cross-lingual

classification, unsupervised and supervised machine translation. On XNLI, our approach pushes the state of the art by an absolute gain of 4.9% accuracy and significantly outperforms approaches presented in section 4.2. Instead of using one LSTM per language, our approach uses a single Transformer for all the XNLI languages and is fine-tuned directly on XNLI English training data. On unsupervised machine translation, we obtain 34.3 BLEU on WMT’16 German-English, improving the previous state of the art by more than 9 BLEU. On supervised machine translation, we obtain a new state of the art of 38.5 BLEU on WMT’16 Romanian-English, outperforming the previous best approach by more than 4 BLEU. Our code and pretrained models are publicly available.⁹

4.3.1 Introduction

As discussed previously, pretraining of sentence encoders with language modeling has led to strong improvements on numerous natural language understanding benchmarks. In this context, a Transformer language model is learned on a large unsupervised text corpus, and then fine-tuned on NLU tasks such as classification or NLI. Research in that area has been essentially monolingual, and largely focused around English benchmarks such as SentEval or GLUE. In this section we extend the idea of language model pretraining to the cross-lingual setting to mitigate the English-centric bias, and build on top of the work presented in section 4.2.

We demonstrate the effectiveness of cross-lingual language model pretraining on multiple cross-lingual understanding benchmarks, including XNLI. Precisely, we make the following contributions:

1. We introduce a new unsupervised method for learning cross-lingual representations using cross-lingual language modeling and investigate two monolingual pretraining objectives.
2. We introduce a new supervised learning objective that improves cross-lingual pretraining when parallel data is available.
3. We significantly outperform the previous state of the art on cross-lingual classification, unsupervised machine translation and supervised machine translation.
4. We show that cross-lingual language models can provide significant improvements on the perplexity of low-resource languages.
5. We make our code and pretrained models publicly available.

⁹<https://github.com/facebookresearch/XLM>

4.3.2 Related Work

While the methods introduced in section 4.2 and the recent work of (Eriguchi et al., 2018; Artetxe and Schwenk, 2018) require a significant amount of parallel data, our recent work in unsupervised machine translation show that sentence representations can be aligned in a completely unsupervised way (Lample et al., 2018a; Artetxe et al., 2018). For instance, in Lample et al. (2018b) we obtained 25.2 BLEU on WMT’16 German-English without using parallel sentences. Similar to this work and the work we presented in section 4.1, we show that we can align distributions of sentences in a completely unsupervised way, and that our cross-lingual models can be used for a broad set of natural language understanding tasks, including machine translation.

The most similar work to ours is probably the one of Wada and Iwata (2018), where the authors train a LSTM (Hochreiter and Schmidhuber, 1997) language model with sentences from different languages. They share the LSTM parameters, but use different lookup tables to represent the words in each language. They focus on aligning word representations and show that their approach work well on word translation tasks.

4.3.3 Cross-lingual language models

In this section, we present the three language modeling objectives we consider throughout this work. Two of them only require monolingual data (unsupervised), while the third one requires parallel sentences (supervised). We consider N languages. Unless stated otherwise, we suppose that we have N monolingual corpora $\{C_i\}_{i=1\dots N}$, and we denote by n_i the number of sentences in C_i .

Shared sub-word vocabulary

In all our experiments we process all languages with the same shared vocabulary created through Byte Pair Encoding (BPE) (Sennrich et al., 2015b). As shown in Lample et al. (2018a), this greatly improves the alignment of embedding spaces across languages that share either the same alphabet or anchor tokens such as digits (Smith et al., 2017) or proper nouns. We learn the BPE splits on the concatenation of sentences sampled randomly from the monolingual corpora. Sentences are sampled according to a multinomial distribution with probabilities $\{q_i\}_{i=1\dots N}$, where:

$$q_i = \frac{p_i^\alpha}{\sum_{j=1}^N p_j^\alpha} \quad \text{with} \quad p_i = \frac{n_i}{\sum_{k=1}^N n_k}.$$

We consider $\alpha = 0.5$. Sampling with this distribution increases the number of tokens associated to low-resource languages and alleviates the bias towards high-resource languages. In particular, this prevents words of low-resource languages from being split at the character level.

Causal Language Modeling (CLM)

Our causal language modeling (CLM) task consists of a Transformer language model trained to model the probability of a word given the previous words in a sentence $P(w_t|w_1, \dots, w_{t-1}, \theta)$. While recurrent neural networks obtain state-of-the-art performance on language modeling benchmarks (Mikolov et al., 2010; Jozefowicz et al., 2016), Transformer models are also very competitive (Dai et al., 2019).

In the case of LSTM language models, back-propagation through time (Werbos, 1990) (BPTT) is performed by providing the LSTM with the last hidden state of the previous iteration. In the case of Transformers, previous hidden states can be passed to the current batch (Al-Rfou et al., 2018) to provide context to the first words in the batch. However, this technique does not scale to the cross-lingual setting, so we just leave the first words in each batch without context for simplicity.

Masked Language Modeling (MLM)

We also consider the masked language modeling (MLM) objective of Devlin et al. (2018), also known as the Cloze task (Taylor, 1953). Following Devlin et al. (2018), we sample randomly 15% of the BPE tokens from the text streams, replace them by a [MASK] token 80% of the time, by a random token 10% of the time, and we keep them unchanged 10% of the time. Differences between our approach and the MLM of Devlin et al. (2018) include the use of text streams of an arbitrary number of sentences (truncated at 256 tokens) instead of pairs of sentences. To counter the imbalance between rare and frequent tokens (e.g. punctuations or stop words), we also subsample the frequent outputs using an approach similar to Mikolov et al. (2013c): tokens in a text stream are sampled according to a multinomial distribution, whose weights are proportional to the square root of their invert frequencies. Our MLM objective is illustrated in Figure 4.6.

Translation Language Modeling (TLM)

Both the CLM and MLM objectives are unsupervised and only require monolingual data. However, these objectives cannot be used to leverage parallel data when it is available. We introduce a new translation language modeling (TLM) objective for improving cross-lingual pretraining. Our

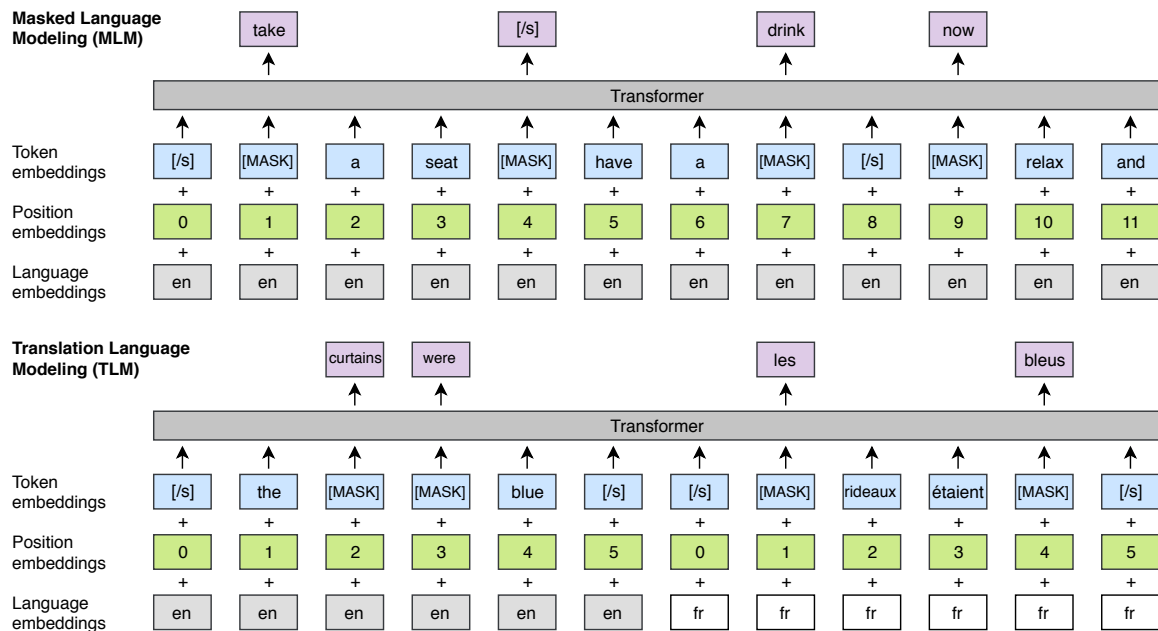


Figure 4.6: **Cross-lingual language model pretraining.** The MLM objective is similar to the one of Devlin et al. (2018), but with continuous streams of text as opposed to sentence pairs. The TLM objective extends MLM to pairs of parallel sentences. To predict a masked English word, the model can attend to both the English sentence and its French translation, and is encouraged to align English and French representations. Position embeddings of the target sentence are reset to facilitate the alignment.

TLM objective is an extension of MLM, where instead of considering monolingual text streams, we concatenate parallel sentences as illustrated in Figure 4.6. We randomly mask words in both the source and target sentences. To predict a word masked in an English sentence, the model can either attend to surrounding English words or to the French translation, encouraging the model to align the English and French representations. In particular, the model can leverage the French context if the English one is not sufficient to infer the masked English words. To facilitate the alignment, we also reset the positions of target sentences.

Cross-lingual Language Models

In this section, we consider cross-lingual language model pretraining with either CLM, MLM, or MLM used in combination with TLM. For the CLM and MLM objectives, we train the model with batches of 64 streams of continuous sentences composed of 256 tokens. At each iteration, a batch is composed of sentences coming from the same language, which is sampled from the distribution

$\{q_i\}_{i=1\dots N}$ above, with $\alpha = 0.7$. When TLM is used in combination with MLM, we alternate between these two objectives, and sample the language pairs with a similar approach.

4.3.4 Cross-lingual language model pretraining

In this section, we explain how cross-lingual language models can be used to obtain:

- a better initialization of sentence encoders for zero-shot cross-lingual classification
- a better initialization of supervised and unsupervised neural machine translation systems
- language models for low-resource languages
- unsupervised cross-lingual word embeddings

Cross-lingual classification

Our pretrained XLM models provide general-purpose cross-lingual text representations. Similar to monolingual language model fine-tuning (Radford et al., 2018; Devlin et al., 2018) on English classification tasks, we fine-tune XLMs on XNLI to evaluate our approach. Precisely, we add a linear classifier on top of the first hidden state of the pretrained Transformer, and fine-tune all parameters on the English NLI training dataset. We then evaluate the capacity of our model to make correct NLI predictions in the 15 XNLI languages. Following our approach in section 4.2, we also include machine translation baselines of train and test sets. We report our results in Table 4.12.

Unsupervised Machine Translation

Pretraining is a key ingredient of unsupervised neural machine translation (UNMT) (Lample et al., 2018a; Artetxe et al., 2018). In Lample et al. (2018b) we show that the quality of pretrained cross-lingual word embeddings used to initialize the lookup table has a significant impact on the performance of an unsupervised machine translation model. In this section, we propose to take this idea one step further by pretraining the entire encoder and decoder with a cross-lingual language model to bootstrap the iterative process of UNMT. We explore various initialization schemes and evaluate their impact on several standard machine translation benchmarks, including WMT'14 English-French, WMT'16 English-German and WMT'16 English-Romanian. Results are presented in Table 4.13.

Supervised Machine Translation

We also investigate the impact of cross-lingual language modeling pretraining for supervised machine translation, and extend the approach of Ramachandran et al. (2016) to multilingual NMT (Johnson et al., 2017b). We evaluate the impact of both CLM and MLM pretraining on WMT’16 Romanian-English, and present results in Table 4.14.

Low-resource language modeling

For low-resource languages, it is often beneficial to leverage data in similar but higher-resource languages, especially when they share a significant fraction of their vocabularies. For instance, there are about 100k sentences written in Nepali on Wikipedia, and about 6 times more in Hindi. These two languages also have more than 80% of their tokens in common in a shared BPE vocabulary of 100k subword units. We provide in Table 4.15 a comparison in perplexity between a Nepali language model and a cross-lingual language model trained in Nepali but enriched with different combinations of Hindi and English data.

Unsupervised cross-lingual word embeddings

In section 4.1, we showed how to perform unsupervised word translation by aligning monolingual word embedding spaces with adversarial training (*MUSE*). In Lample et al. (2018a), we showed that using a shared vocabulary between two languages and then applying fastText (Bojanowski et al., 2017) on the concatenation of their monolingual corpora also directly provides high-quality cross-lingual word embeddings (*Concat*) for languages that share a common alphabet. In this section, we also use a shared vocabulary but our word embeddings are obtained via the lookup table of our cross-lingual language model (*XLM*). In Section 4.3.5, we compare these three approaches on three different metrics: cosine similarity, L2 distance and cross-lingual word similarity.

4.3.5 Experiments and results

In this section, we empirically demonstrate the strong impact of cross-lingual language model pretraining on several benchmarks, and compare our approach to the current state of the art.

Training details

In all experiments, we use a Transformer architecture with 1024 hidden units, 8 heads, GELU activations (Hendrycks and Gimpel, 2016), a dropout rate of 0.1 and learned positional embeddings. We train our models with the Adam optimizer (Kingma and Ba, 2014), a linear warm-up (Vaswani et al., 2017) and learning rates varying from 10^{-4} to $5 \cdot 10^{-4}$.

For the CLM and MLM objectives, we use streams of 256 tokens and a mini-batches of size 64. Unlike Devlin et al. (2018), a sequence in a mini-batch can contain more than two consecutive sentences, as explained in Section 4.3.3. For the TLM objective, we sample mini-batches of 4000 tokens composed of sentences with similar lengths. We use the averaged perplexity over languages as a stopping criterion for training. For machine translation, we only use 6 layers, and we create mini-batches of 2000 tokens.

When fine-tuning on XNLI, we use mini-batches of size 8 or 16, and we clip the sentence length to 256 words. We use 80k BPE splits and a vocabulary of 95k and train a 12-layer model on the Wikipedias of the XNLI languages. We sample the learning rate of the Adam optimizer with values from $5 \cdot 10^{-4}$ to $2 \cdot 10^{-4}$, and use small evaluation epochs of 20000 random samples. We use the first hidden state of the last layer of the transformer as input to the randomly initialized final linear classifier, and fine-tune all parameters. In our experiments, using either max-pooling or mean-pooling over the last layer did not work better than using the first hidden state.

We implement all our models in PyTorch (Paszke et al., 2017), and train them on 64 Volta GPUs for the language modeling tasks, and 8 GPUs for the MT tasks. We use float16 operations to speed up training and to reduce the memory usage of our models.

Data preprocessing

We use *WikiExtractor*¹⁰ to extract raw sentences from Wikipedia dumps and use them as monolingual data for the CLM and MLM objectives. For the TLM objective, we only use parallel data that involves English, as we did in section 4.2 for our XNLI experiments. Precisely, we use MultiUN (Ziemski et al., 2016) for French, Spanish, Russian, Arabic and Chinese, and the IIT Bombay corpus (Anoop et al., 2018) for Hindi. We extract the following corpora from the OPUS¹¹ website (Tiedemann, 2012): the EUbookshop corpus for German, Greek and Bulgarian, OpenSubtitles 2018 for Turkish, Vietnamese and Thai, Tanzil for both Urdu and Swahili and GlobalVoices for Swahili. For

¹⁰<https://github.com/attardi/wikiextractor>

¹¹<http://opus.nlpl.eu>

	en	fr	es	de	el	bg	ru	tr	ar	vi	th	zh	hi	sw	ur	Δ
<i>Machine translation baselines (TRANSLATE-TRAIN)</i>																
Devlin et al. (2018)	81.9	-	77.8	75.9	-	-	-	-	70.7	-	-	76.6	-	-	61.6	-
XLM (MLM+TLM)	<u>85.0</u>	<u>80.2</u>	<u>80.8</u>	<u>80.3</u>	<u>78.1</u>	<u>79.3</u>	<u>78.1</u>	<u>74.7</u>	<u>76.5</u>	<u>76.6</u>	<u>75.5</u>	<u>78.6</u>	<u>72.3</u>	<u>70.9</u>	63.2	<u>76.7</u>
<i>Machine translation baselines (TRANSLATE-TEST)</i>																
Devlin et al. (2018)	81.4	-	74.9	74.4	-	-	-	-	70.4	-	-	70.1	-	-	62.1	-
XLM (MLM+TLM)	<u>85.0</u>	79.0	79.5	78.1	77.8	77.6	75.5	73.7	73.7	70.8	70.4	73.6	69.0	64.7	65.1	74.2
<i>Evaluation of cross-lingual sentence encoders</i>																
Conneau et al. (2018c)	73.7	67.7	68.7	67.7	68.9	67.9	65.4	64.2	64.8	66.4	64.1	65.8	64.1	55.7	58.4	65.6
Devlin et al. (2018)	81.4	-	74.3	70.5	-	-	-	-	62.1	-	-	63.8	-	-	58.3	-
Artetxe and Schwenk (2018)	73.9	71.9	72.9	72.6	73.1	74.2	71.5	69.7	71.4	72.0	69.2	71.4	65.5	62.2	61.0	70.2
XLM (MLM)	83.2	76.5	76.3	74.2	73.1	74.0	73.1	67.8	68.5	71.2	69.2	71.9	65.7	64.6	63.4	71.5
XLM (MLM+TLM)	<u>85.0</u>	<u>78.7</u>	<u>78.9</u>	<u>77.8</u>	<u>76.6</u>	<u>77.4</u>	<u>75.3</u>	<u>72.5</u>	<u>73.1</u>	<u>76.1</u>	<u>73.2</u>	<u>76.5</u>	<u>69.6</u>	<u>68.4</u>	<u>67.3</u>	<u>75.1</u>

Table 4.12: **Results on cross-lingual classification accuracy.** Test accuracy on the 15 XNLI languages. We report results for machine translation baselines and zero-shot classification approaches based on cross-lingual sentence encoders. XLM (MLM) corresponds to our unsupervised approach trained only on monolingual corpora, and XLM (MLM+TLM) corresponds to our supervised method that leverages both monolingual and parallel data through the TLM objective. Δ corresponds to the average accuracy.

Chinese, Japanese and Thai we use the tokenizer of Chang et al. (2008), the *Kytea*¹² tokenizer, and the *PyThaiNLP*¹³ tokenizer respectively. For all other languages, we use the tokenizer provided by Moses (Koehn et al., 2007), falling back on the default English tokenizer when necessary. We use fastBPE¹⁴ to learn BPE codes and split words into subword units. The BPE codes are learned on the concatenation of sentences sampled from all languages, following the method presented in Section 4.3.3.

Results and analysis

In the following, we demonstrate the effectiveness of cross-lingual language model pretraining. Our approach significantly outperforms the previous state of the art on cross-lingual classification, unsupervised and supervised machine translation.

Cross-lingual classification In Table 4.12, we evaluate two types of pretrained cross-lingual encoders: an unsupervised cross-lingual language model that uses the MLM objective on monolingual corpora only; and a supervised cross-lingual language model that combines both the MLM

¹²<http://www.phontron.com/kytea>

¹³<https://github.com/PyThaiNLP/pythainlp>

¹⁴<https://github.com/glample/fastBPE>

and the TLM loss using additional parallel data. Following 4.2, we include two machine translation baselines: TRANSLATE-TRAIN, where the English MultiNLI training set is machine translated into each XNLI language, and TRANSLATE-TEST where every dev and test set of XNLI is translated to English. We report the XNLI baselines of section 4.2, the multilingual BERT approach of Devlin et al. (2018) and the recent work of Artetxe and Schwenk (2018).

Our fully unsupervised MLM method sets a new state of the art on zero-shot cross-lingual classification and significantly outperforms the supervised approach of Artetxe and Schwenk (2018) which uses 223 million of parallel sentences. Precisely, MLM obtains 71.5% accuracy on average (Δ), while they obtained 70.2% accuracy. By leveraging parallel data through the TLM objective (MLM+TLM), we get a significant boost in performance of 3.6% accuracy, improving even further the state of the art to 75.1%. On the Swahili and Urdu low-resource languages, we outperform the previous state of the art by 6.2% and 6.3% respectively. Due to additional English data in parallel corpora, we observe that TLM, in addition to MLM, also improves English accuracy from 83.2% to 85% accuracy, outperforming Artetxe and Schwenk (2018) and Devlin et al. (2018) by 11.1% and 3.6% accuracy respectively.

When fine-tuned on the training set of each XNLI language (TRANSLATE-TRAIN), our supervised model outperforms our zero-shot approach by 1.6%, reaching an absolute state of the art of 76.7% average accuracy. This result demonstrates in particular the consistency of our approach and shows that XLMs can be fine-tuned on any language with strong performance. Similar to the multilingual BERT (Devlin et al., 2018), we observe that TRANSLATE-TRAIN outperforms TRANSLATE-TEST by 2.5% average accuracy, and additionally that our zero-shot approach outperforms TRANSLATE-TEST by 0.9%.

Unsupervised machine translation For the unsupervised machine translation task we consider three language pairs: English-French, English-German, and English-Romanian. Our setting is identical to the one of Lample et al. (2018b), except for the initialization step where we use cross-lingual language modeling to pretrain the full model as opposed to only the lookup table.

For both the encoder and the decoder, we consider different possible initializations: CLM pre-training, MLM pretraining, or random initialization, which results in nine different settings. We then follow Lample et al. (2018b) and train the model with a denoising auto-encoding loss along with an online back-translation loss. Results are reported in Table 4.13. We compare our approach with the ones of Lample et al. (2018b). For each language pair, we observe significant improvements over the previous state of the art. We re-implemented the NMT approach of Lample et al. (2018b) (EMB),

		en-fr	fr-en	en-de	de-en	en-ro	ro-en
<i>Previous state-of-the-art - Lample et al. (2018b)</i>							
NMT		25.1	24.2	17.2	21.0	21.2	19.4
PBSMT		28.1	27.2	17.8	22.7	21.3	23.0
PBSMT + NMT		27.6	27.7	20.2	25.2	25.1	23.9
<i>Our results for different encoder and decoder initializations</i>							
EMB	EMB	29.4	29.4	21.3	27.3	27.5	26.6
-	-	13.0	15.8	6.7	15.3	18.9	18.3
-	CLM	25.3	26.4	19.2	26.0	25.7	24.6
-	MLM	29.2	29.1	21.6	28.6	28.2	27.3
CLM	-	28.7	28.2	24.4	30.3	29.2	28.0
CLM	CLM	30.4	30.0	22.7	30.5	29.0	27.8
CLM	MLM	32.3	31.6	24.3	32.5	31.6	29.8
MLM	-	31.6	32.1	27.0	33.2	31.8	30.5
MLM	CLM	33.4	32.3	24.9	32.9	31.7	30.4
MLM	MLM	33.4	33.3	26.4	34.3	33.3	31.8

Table 4.13: **Results on unsupervised MT.** BLEU scores on WMT’14 English-French, WMT’16 German-English and WMT’16 Romanian-English. For our results, the first two columns indicate the model used to pretrain the encoder and the decoder. “ - ” means the model was randomly initialized. EMB corresponds our reimplementation of the original pretraining of the lookup table with cross-lingual embeddings from Lample et al. (2018a), CLM and MLM correspond to pretraining with models trained on the CLM or MLM objectives.

and obtained better results than reported in their paper. We expect that this is due to our multi-GPU implementation which uses significantly larger batches. In German-English, our best model outperforms the previous unsupervised approach by more than 9.1 BLEU, and 13.3 BLEU if we only consider neural unsupervised approaches. Compared to pretraining only the lookup table (EMB), pretraining both the encoder and decoder with MLM leads to consistent significant improvements of up to 7 BLEU on German-English. We also observe that the MLM objective pretraining consistently outperforms the CLM one, going from 30.4 to 33.4 BLEU on English-French, and from 28.0 to 31.8 on Romanian-English. These results are consistent with the ones of Devlin et al. (2018) who observed a better generalization on NLU tasks when training on the MLM objective compared to CLM. We also observe that the encoder is the most important element to pretrain: when compared to pretraining both the encoder and the decoder, pretraining only the decoder leads to a significant drop in performance, while pretraining only the encoder only has a small impact on the final BLEU

score.

Pretraining	-	CLM	MLM
Sennrich et al. (2016)	33.9	-	-
ro \rightarrow en	28.4	31.5	35.3
ro \leftrightarrow en	28.5	31.5	35.6
ro \leftrightarrow en + BT	34.4	37.0	38.5

Table 4.14: **Results on supervised MT.** BLEU scores on WMT’16 Romanian-English. The previous state-of-the-art of Sennrich et al. (2016) uses both back-translation and an ensemble model. ro \leftrightarrow en corresponds to models trained on both directions.

Supervised machine translation In Table 4.14 we report the performance on Romanian-English WMT’16 for different supervised training configurations: mono-directional (ro \rightarrow en), bidirectional (ro \leftrightarrow en, a multi-NMT model trained on both en \rightarrow ro and ro \rightarrow en) and bidirectional with back-translation (ro \leftrightarrow en + BT). Models with back-translation are trained with the same monolingual data as language models used for pretraining. As in the unsupervised setting, we observe that pretraining provides a significant boost in BLEU score for each configuration, and that pretraining with the MLM objective leads to the best performance. Also, while models with back-translation have access to the same amount of monolingual data as the pretrained models, they are not able to generalize as well on the evaluation sets. Our bidirectional model trained with back-translation obtains the best performance and reaches 38.5 BLEU, outperforming the previous SOTA of Sennrich et al. (2016) (based on back-translation and ensemble models) by more than 4 BLEU.

Low-resource language model In Table 4.15, we investigate the impact of cross-lingual language modeling for improving the perplexity of a Nepali language model. To do so, we train a Nepali language model on Wikipedia, together with additional data from either English or Hindi. While Nepali and English are distant languages, Nepali and Hindi are similar as they share the same Devanagari script and have a common Sanskrit ancestor. When using English data, we reduce the perplexity on the Nepali language model by 17.1 points, going from 157.2 for Nepali-only language modeling to 140.1 when using English. Using additional data from Hindi, we get a much larger perplexity reduction of 41.6. Finally, by leveraging data from both English and Hindi, we reduce the perplexity even more to 109.3 on Nepali. The gains in perplexity from cross-lingual language modeling can be partly explained by the n-grams anchor points that are shared across languages,

for instance in Wikipedia articles. The cross-lingual language model can thus transfer the additional context provided by the Hindi or English monolingual corpora through these anchor points to improve the Nepali language model.

Training languages	Nepali perplexity
Nepali	157.2
Nepali + English	140.1
Nepali + Hindi	115.6
Nepali + English + Hindi	109.3

Table 4.15: **Results on language modeling.** Nepali perplexity when using additional data from a similar language (Hindi) or a distant one (English).

Unsupervised cross-lingual word embeddings The MUSE, Concat and XLM (MLM) methods provide unsupervised cross-lingual word embedding spaces that have different properties. In Table 4.16, we study those three methods using the same word vocabulary and compute the cosine similarity and L2 distance between word translation pairs from the MUSE dictionaries. We also evaluate the quality of the cosine similarity measure via the SemEval’17 cross-lingual word similarity task of Camacho-Collados et al. (2017). We observe that XLM outperforms both MUSE and Concat on cross-lingual word similarity, reaching a Pearson correlation of 0.69. Interestingly, word translation pairs are also far closer in the XLM cross-lingual word embedding space than for MUSE or Concat. Specifically, MUSE obtains 0.38 and 5.13 for cosine similarity and L2 distance while XLM gives 0.55 and 2.64 for the same metrics. Note that XLM embeddings have the particularity of being trained together with a sentence encoder which may enforce this closeness, while MUSE and Concat are based on fastText word embeddings.

	Cosine sim.	L2 dist.	SemEval’17
MUSE	0.38	5.13	0.65
Concat	0.36	4.89	0.52
XLM	0.55	2.64	0.69

Table 4.16: **Unsupervised cross-lingual word embeddings** Cosine similarity and L2 distance between source words and their translations. Pearson correlation on SemEval’17 cross-lingual word similarity task of Camacho-Collados et al. (2017).

4.3.6 Conclusion

In this section, we show for the first time the strong impact of cross-lingual language model (XLM) pretraining. We investigate two unsupervised training objectives that require only monolingual corpora: Causal Language Modeling (CLM) and Masked Language Modeling (MLM). We show that both the CLM and MLM approaches provide strong cross-lingual features that can be used for pretraining models. On unsupervised machine translation, we show that MLM pretraining is extremely effective. We reach a new state of the art of 34.3 BLEU on WMT’16 German-English, outperforming the previous best unsupervised approach by more than 9 BLEU. Similarly, we obtain strong improvements on supervised machine translation. We reach a new state of the art on WMT’16 Romanian-English of 38.5 BLEU, which corresponds to an improvement of more than 4 BLEU points. We also demonstrate that cross-lingual language model can be used to improve the perplexity of a Nepali language model, and that it provides unsupervised cross-lingual word embeddings. Without using a single parallel sentence, a cross-lingual language model fine-tuned on the XNLI cross-lingual classification benchmark already outperforms the previous supervised state of the art by 1.3% accuracy on average. A key contribution of our work is the translation language modeling (TLM) objective which improves cross-lingual language model pretraining by leveraging parallel data. TLM naturally extends the BERT MLM approach by using batches of parallel sentences instead of consecutive sentences. We obtain a significant gain by using TLM in addition to MLM, and we show that this supervised approach beats the previous state of the art on XNLI by 4.9% accuracy on average. Our code and pretrained models are publicly available.

Chapter 5

Conclusion

In this thesis, we proposed new methods to learn, evaluate, analyze and align sentence embedding spaces. First in the monolingual setting, we proposed SentEval (Conneau and Kiela, 2018; Conneau et al., 2018a), a toolkit to evaluate and analyze fixed-size sentence embeddings. We conducted a thorough analysis on the impact of the encoder architecture and the training task on the embedding space, using both downstream and probing tasks. At the time of publication, we showed that In-ferSent (Conneau et al., 2017) obtained state-of-the-art results and outperformed SkipThought and bag-of-vectors. Our contributions inspired other researchers which extended our approach using multi-task learning (Subramanian et al., 2018; Cer et al., 2018) and outperformed our approach. In parallel, a new line of work emerged on sentence encoder fine-tuning where instead of learning a logistic regression or a multi-layer perceptron on top of fixed-size frozen representations, encoders were fine-tuned entirely on the downstream tasks. These models were evaluated on the GLUE benchmark (Wang et al., 2018). The research community rediscovered the power of language models as a way to learn powerful sentence representations (Peters et al., 2018; Howard and Ruder, 2018; Radford et al., 2018; Devlin et al., 2018), and showed strong improvements on a number of NLU tasks.

Both areas of research on frozen fixed-size embeddings and encoder fine-tuning were very English-centric, in particular because SentEval and GLUE only incorporate English evaluation benchmarks. In this thesis, we extended these approaches to the cross-lingual world and built cross-lingual embedding spaces. We started by aligning word representations in Conneau et al. (2018b) without parallel data using adversarial training (Ganin et al., 2016). We showed that we could translate words between various languages while having only access to monolingual corpora.

We switched from word embeddings alignment to sentence embeddings alignment by first proving the community with the first large-scale cross-lingual classification benchmark called XNLI in order to catalyze research in cross-lingual understanding. We showed that we could align sentence embedding spaces using an alignment loss similar to a ranking loss, such that while having seen only English NLI training data, we are still able to make predictions in the 14 other languages of XNLI. Finally, we explored the power of cross-lingual language models as a way to learn strong multilingual representations of sentences and applied to a series of XLU tasks. In particular we showed strong improvements for supervised and unsupervised machine translation by pretraining both the encoder and the decoder using a cross-lingual language model. We also showed that we could leverage additional data from Hindi to improve a Nepalese language model and that we could learn a cross-lingual sentence encoder without any parallel data. We improved our cross-lingual masked language model by leveraging available parallel data using a new objective called translation language model that extend the BERT approach to the cross-lingual setting. We obtained strong improvements on XNLI and outperformed the previous state of the art by 4.9% accuracy on average.

In the recent years, we have seen a growing interest in multi-task learning and multilinguality. With strong evaluation benchmarks like XNLI, there is indeed a lot of opportunity to build better systems for low-resource languages. In particular, a lot of work has to be done in the direction of cross-lingual language modeling for scaling to multiple languages and larger datasets, or improving alignment techniques beyond translation language modeling. Leveraging redundancy in Wikipedia-based question answering system is also an interesting research direction that we hope to explore in the near future. We believe our work only scratches the surface of what is possible to do in cross-lingual understanding with neural networks and we hope the research community will build even stronger NLP systems for low-resource languages.

Bibliography

- Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2017. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. In *Proceedings of ICLR Conference Track*. Toulon, France. Published online: <https://openreview.net/group?id=ICLR.cc/2017/conference>.
- Željko Agić and Natalie Schluter. 2018. Baselines and test data for cross-lingual inference. *LREC*.
- Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca, and Aitor Soroa. 2009. A study on similarity and relatedness using distributional and WordNet-based approaches. In *Proceedings of HLT-NAACL*. Boulder, CO, pages 19–27.
- Eneko Agirre, Carmen Banea, Claire Cardie, Daniel M Cer, Mona T Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Inigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, et al. 2015. Semeval-2015 task 2: Semantic textual similarity, english, spanish and pilot on interpretability. In *SemEval@NAACL-HLT*. pages 252–263.
- Eneko Agirre, Carmen Baneab, Claire Cardie, Daniel Cerd, Mona Diabe, Aitor Gonzalez-Agirre, Weiwei Guof, Rada Mihalceab, German Rigaua, and Janyce Wiebeg. 2014. Semeval-2014 task 10: Multilingual semantic textual similarity. *SemEval 2014* page 81.
- Eneko Agirre, Carmen Baneab, Daniel Cerd, Mona Diabe, Aitor Gonzalez-Agirre, Rada Mihalceab, German Rigaua, Janyce Wiebef, and Basque Country Donostia. 2016. Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. *Proceedings of SemEval* pages 497–511.
- Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-agirre, and Weiwei Guo. 2013. sem 2013 shared task: Semantic textual similarity, including a pilot on typed-similarity. In *In *SEM 2013:*

- The Second Joint Conference on Lexical and Computational Semantics. Association for Computational Linguistics.*
- Eneko Agirre, Mona Diab, Daniel Cer, and Aitor Gonzalez-Agirre. 2012. Semeval-2012 task 6: A pilot on semantic textual similarity. In *Proceedings of Semeval-2012*. pages 385–393.
- Rami Al-Rfou, Dokook Choe, Noah Constant, Mandy Guo, and Llion Jones. 2018. Character-level language modeling with deeper self-attention. *arXiv preprint arXiv:1808.04444* .
- Waleed Ammar, George Mulcaire, Yulia Tsvetkov, Guillaume Lample, Chris Dyer, and Noah A Smith. 2016. Massively multilingual word embeddings. *arXiv preprint arXiv:1602.01925* .
- Kunchukuttan Anoop, Mehta Pratik, and Bhattacharyya Pushpak. 2018. The iit bombay english-hindi parallel corpus. In *LREC*.
- Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*. pages 2425–2433.
- Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. *International Conference on Learning Representations* .
- Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2016. Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. *Proceedings of EMNLP* .
- Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2017. Learning bilingual word embeddings with (almost) no bilingual data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. volume 1, pages 451–462.
- Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2018. Unsupervised neural machine translation. In *ICLR*.
- Mikel Artetxe and Holger Schwenk. 2018. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *arXiv preprint arXiv:1812.10464* .
- Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *Advances in neural information processing systems (NIPS)* .

- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of ICLR Conference Track*. San Diego, CA. Published online: <http://www.iclr.cc/doku.php?id=iclr2015:main>.
- Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. 2009. The wacky wide web: a collection of very large linguistically processed web-crawled corpora. *Language resources and evaluation* 43(3):209–226.
- Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2017a. What do neural machine translation models learn about morphology? In *Proceedings of ACL*. Vancouver, Canada, pages 861–872.
- Yonatan Belinkov, Lluís Màrquez, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James Glass. 2017b. Evaluating layers of representation in neural machine translation on part-of-speech and semantic tagging tasks. In *Proceedings of IJCNLP*. Taipei, Taiwan, pages 1–10.
- Yoshua Bengio, Rejean Ducharme, and Pascal Vincent. 2003. A neural probabilistic language model. *Journal of Machine Learning Research* 3:1137–1155.
- Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics* 5:135–146.
- Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of EMNLP*.
- Alexander Budanitsky and Graeme Hirst. 2006. Evaluating wordnet-based measures of lexical semantic relatedness. *Computational Linguistics* 32(1):13–47.
- Jose Camacho-Collados, Mohammad Taher Pilehvar, Nigel Collier, and Roberto Navigli. 2017. Semeval-2017 task 2: Multilingual and cross-lingual semantic word similarity. *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval 2017)*.
- José Camacho-Collados, Mohammad Taher Pilehvar, and Roberto Navigli. 2016. Nasari: Integrating explicit knowledge and corpus statistics for a multilingual representation of concepts and entities. *Artificial Intelligence* 240:36–64.

- Hailong Cao, Tiejun Zhao, Shu Zhang, and Yao Meng. 2016. A distribution-based model to learn bilingual word embeddings. *Proceedings of COLING* .
- Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv preprint arXiv:1708.00055* .
- Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175* .
- Sarath Chandar, Mitesh M. Khapra, Balaraman Ravindran, Vikas Raykar, and Amrita Saha. 2013. Multilingual deep learning. In *NIPS, Workshop Track*.
- Pi-Chuan Chang, Michel Galley, and Christopher D Manning. 2008. Optimizing chinese word segmentation for machine translation performance. In *Proceedings of the third workshop on statistical machine translation*. pages 224–232.
- Billy Chiu, Anna Korhonen, and Sampo Pyysalo. 2016. Intrinsic evaluation of word vectors fails to predict extrinsic performance. In *First Workshop on Evaluating Vector Space Representations for NLP (RepEval)*.
- Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014a. On the properties of neural machine translation: Encoder-decoder approaches. In *Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST-8)*.
- Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014b. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *Proceedings of EMNLP* .
- Moustapha Cisse, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. 2017. Parseval networks: Improving robustness to adversarial examples. *International Conference on Machine Learning* pages 854–863.
- Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*. ACM, pages 160–167.

- Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research* 12(Aug):2493–2537.
- Alexis Conneau and Douwe Kiela. 2018. Senteval: An evaluation toolkit for universal sentence representations. *Proceedings of LREC* .
- Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680.
- Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018a. What you can cram into a single vector: Probing sentence embeddings for linguistic properties. In *ACL*.
- Alexis Conneau, Guillaume Lample, Marc’Aurelio Ranzato, Ludovic Denoyer, and Hervé Jegou. 2018b. Word translation without parallel data. In *ICLR*.
- Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel R. Bowman, Holger Schwenk, and Veselin Stoyanov. 2018c. Xnli: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.
- Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann Lecun. 2016. Very deep convolutional networks for text classification. *EACL* .
- Zihang Dai, Zhilin Yang, Yiming Yang, William W. Cohen, Jaime Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Language modeling with longer-term dependency.
- Fahim Dalvi, Nadir Durrani, Hassan Sajjad, Yonatan Belinkov, and Stephan Vogel. 2017. Understanding and improving morphological learning in the neural machine translation decoder. In *Proceedings of IJCNLP*. Taipei, Taiwan, pages 142–151.
- Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. 2016. Language modeling with gated convolutional networks. *arXiv preprint arXiv:1612.08083* .

- Manaal Faruqui, Yulia Tsvetkov, Pushpendre Rastogi, and Chris Dyer. 2016. Problems with evaluation of word embeddings using word similarity tasks. *arXiv preprint arXiv:1605.02276* .
- Kunihiko Fukushima. 1980. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics* 36(4):193–202.
- Pascale Fung. 1995. Compiling bilingual lexicon entries from a non-parallel english-chinese corpus. In *Proceedings of the Third Workshop on Very Large Corpora*. pages 173–183.
- Pascale Fung and Lo Yuen Yee. 1998. An ir approach for translating new words from nonparallel, comparable texts. In *Proceedings of the 17th International Conference on Computational Linguistics - Volume 1*. Association for Computational Linguistics, COLING '98, pages 414–420.
- Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *Journal of Machine Learning Research* 17(59):1–35.
- Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: The paraphrase database. In *Proceedings of NAACL-HLT*. Atlanta, Georgia, pages 758–764.
- Jonas Gehring, Michael Auli, David Grangier, and Yann N Dauphin. 2016. A convolutional encoder model for neural machine translation. *arXiv preprint arXiv:1611.02344* .
- Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of ICML*. Sydney, Australia, pages 1243–1252.
- Yichen Gong, Heng Luo, and Jian Zhang. 2018. Natural language inference over interaction space. *ICLR* .
- Ian Goodfellow. 2016. Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160* .
- Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems* pages 2672–2680.
- S. Gouews, Y. Bengio, and G. Corrado. 2014. Bilbowa: Fast bilingual distributed representations without word alignments. In *ICML*.

- Stephan Gouws, Yoshua Bengio, and Greg Corrado. 2015. Bilbowa: Fast bilingual distributed representations without word alignments. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*. pages 748–756.
- Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. *LREC* .
- Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R Bowman, and Noah A Smith. 2018. Annotation artifacts in natural language inference data. *NAACL* .
- Aria Haghighi, Percy Liang, Taylor Berg-Kirkpatrick, and Dan Klein. 2008. Learning bilingual lexicons from monolingual corpora. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*.
- Zellig S Harris. 1954. Distributional structure. *Word* 10(2-3):146–162.
- Vasileios Hatzivassiloglou, Judith L. Klavans, Melissa L. Holcombe, Regina Barzilay, Min yen Kan, and Kathleen R. McKeown. 2001. SIMFINDER: A Flexible Clustering Tool for Summarization. In *In Proceedings of the NAACL Workshop on Automatic Summarization*. pages 41–49.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of CVPR*.
- Dan Hendrycks and Kevin Gimpel. 2016. Bridging nonlinearities and stochastic regularizers with gaussian error linear units. *arXiv preprint arXiv:1606.08415* .
- Karl Moritz Hermann and Phil Blunsom. 2014. Multilingual models for compositional distributed semantics. In *ACL*. pages 58–68.
- Felix Hill, Kyunghyun Cho, and Anna Korhonen. 2016a. Learning distributed representations of sentences from unlabelled data. *arXiv preprint arXiv:1602.03483* .
- Felix Hill, Roi Reichart, and Anna Korhonen. 2016b. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics* .
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

- Jeremy Howard and Sebastian Ruder. 2018. Fine-tuned language models for text classification. In *ACL*.
- Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of SIGKDD*, pages 168–177.
- Dieuwke Hupkes, Sara Veldhoen, and Willem Zuidema. 2017. Visualisation and diagnostic classifiers reveal how recurrent and recursive neural networks process hierarchical structure. <http://arxiv.org/abs/1711.10203>.
- Ann Irvine and Chris Callison-Burch. 2013. Supervised bilingual lexicon induction with multiple monolingual signals. In *HLT-NAACL*.
- Allan Jabri, Armand Joulin, and Laurens van der Maaten. 2016. Revisiting visual question answering baselines. In *Proceedings of ECCV*. Amsterdam, the Netherlands, pages 727–739.
- Herve Jegou, Cordelia Schmid, Hedi Harzallah, and Jakob Verbeek. 2010. Accurate image search using the contextual dissimilarity measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32(1):2–11.
- Yangfeng Ji and Jacob Eisenstein. 2013. Discriminative improvements to distributional sentence similarity. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
- Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2017a. Billion-scale similarity search with gpus. *arXiv preprint arXiv:1702.08734* .
- Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2017b. Google’s multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics* 5:339–351.
- Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759* .
- Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. 2016. Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410* .

- Àkos Kádár, Grzegorz Chrupała, and Afra Alishahi. 2017. Representation of linguistic form and function in recurrent neural networks. *Computational Linguistics* 43(4):761–780.
- Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. *ACL*.
- Andrej Karpathy and Li Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of CVPR*, pages 3128–3137.
- Douwe Kiela, Alexis Conneau, Allan Jabri, and Maximilian Nickel. 2017. Learning visually grounded sentence representations. *Proceedings of NAACL 2018*.
- Yoon Kim. 2014. Convolutional neural networks for sentence classification. *EMNLP*.
- Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*.
- Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in neural information processing systems*, pages 3294–3302.
- Dan Klein and Christopher Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of ACL*, Sapporo, Japan, pages 423–430.
- Alexandre Klementiev, Ivan Titov, and Binod Bhattarai. 2012. Inducing crosslingual distributed representations of words. *Proceedings of COLING 2012* pages 1459–1474.
- T. Kociský, K.M. Hermann, and P. Blunsom. 2014. Learning bilingual word representations by marginalizing alignments. In *ACL*, pages 224–229.
- Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86.
- Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*. Association for Computational Linguistics, pages 177–180.

- Philipp Koehn and Kevin Knight. 2002. Learning a translation lexicon from monolingual corpora. In *Proceedings of the ACL-02 workshop on Unsupervised lexical acquisition-Volume 9*. Association for Computational Linguistics, pages 9–16.
- Grzegorz Kondrak, Bradley Hauer, and Garrett Nicolai. 2017. Bootstrapping unsupervised bilingual lexicon induction. In *EACL*.
- Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*. pages 1097–1105.
- Alice Lai and Julia Hockenmaier. 2014. Illinois-lh: A denotational and distributional approach to semantics. *Proc. SemEval 2:5*.
- Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of NAACL-HLT*. pages 260–270.
- Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*.
- Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc’Aurelio Ranzato. 2018a. Unsupervised machine translation using monolingual corpora only. In *ICLR*.
- Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc’Aurelio Ranzato. 2018b. Phrase-based & neural unsupervised machine translation. In *EMNLP*.
- Angeliki Lazaridou, Georgiana Dinu, and Marco Baroni. 2015. Hubness and pollution: Delving into cross-space mapping for zero-shot learning. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*.
- Quoc V Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *ICML*. volume 14, pages 1188–1196.
- Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11):2278–2324.
- Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. *Advances in neural information processing systems* pages 2177–2185.

- Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. 2016. Visualizing and understanding neural models in NLP. In *Proceedings of NAACL*. San Diego, CA, pages 681–691.
- Jiwei Li, Monroe Will, and Dan Jurafsky. 2017. Understanding neural networks through representation erasure. <https://arxiv.org/abs/1612.08220>.
- Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*. pages 740–755.
- Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. *International Conference on Learning Representations* .
- Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics* 4:521–535.
- Etai Littwin and Lior Wolf. 2016. The multiverse loss for robust transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pages 3957–3966.
- Yang Liu, Chengjie Sun, Lei Lin, and Xiaolong Wang. 2016. Learning natural language inference using bidirectional lstm model and inner-attention. *arXiv preprint arXiv:1605.09090* .
- Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2015. Multi-task sequence to sequence learning. *arXiv preprint arXiv:1511.06114* .
- Thang Luong, Richard Socher, and Christopher D Manning. 2013. Better word representations with recursive neural networks for morphology. *CoNLL* pages 104–113.
- Lin Ma, Zhengdong Lu, Lifeng Shang, and Hang Li. 2015. Multimodal convolutional neural networks for matching image and sentence. In *Proceedings of the IEEE International Conference on Computer Vision*. pages 2623–2631.
- Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A sick cure for the evaluation of compositional distributional semantic models. In *Proceedings of LREC*. pages 216–223.

- Yashar Mehdad, Matteo Negri, and Marcello Federico. 2011. Using bilingual parallel corpora for cross-lingual textual entailment. In *ACL*. pages 1336–1345.
- Tomas Mikolov. 2012. *Statistical language models based on neural networks*. Dissertation, Brno University of Technology.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *Proceedings of Workshop at ICLR* .
- Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhersch, and Armand Joulin. 2017. Advances in pre-training distributed word representations .
- Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černocký, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Eleventh Annual Conference of the International Speech Communication Association*.
- Tomas Mikolov, Quoc V Le, and Ilya Sutskever. 2013b. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168* .
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013c. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. pages 3111–3119.
- Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013d. Linguistic regularities in continuous space word representations. In *Proceedings of NAACL*. Atlanta, Georgia, pages 746–751.
- Saif M. Mohammad, Mohammad Salameh, and Svetlana Kiritchenko. 2016. How translation alters sentiment. *J. Artif. Int. Res.* 55(1):95–130.
- Lili Mou, Zhao Meng, Rui Yan, Ge Li, Yan Xu, Lu Zhang, and Zhi Jin. 2016. How transferable are neural networks in nlp applications? *arXiv preprint arXiv:1603.06111* .
- Tasha Nagamine, Michael L. Seltzer, and Nima Mesgarani. 2015. Exploring how deep neural networks form phonemic categories. In *Proceedings of INTERSPEECH*. Dresden, Germany, pages 1912–1916.
- Neha Nayak, Gabor Angeli, and Christopher D Manning. 2016. Evaluating word embeddings using a representative suite of practical tasks. In *First Workshop on Evaluating Vector Space Representations for NLP (RepEval)*. page 19.

- Matteo Negri, Luisa Bentivogli, Yashar Mehdad, Danilo Giampiccolo, and Alessandro Marchetti. 2011. Divide and conquer: Crowdsourcing the creation of cross-lingual textual entailment corpora. In *EMNLP*. pages 670–679.
- Matthew Nelson, Imen El Karoui, Kristof Giber, Xiaofang Yang, Laurent Cohen, Hilda Koopman, Sydney Cash, Lionel Naccache, John Hale, Christophe Pallier, and Stanislas Dehaene. 2017. Neurophysiological dynamics of phrase-structure building during sentence processing. *Proceedings of the National Academy of Sciences* 114(18):E3669–E3678.
- Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. 2014. Learning and transferring mid-level image representations using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. pages 1717–1724.
- Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of ACL*. page 271.
- Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of ACL*. pages 115–124.
- Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernandez. 2016. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of ACL*. Berlin, Germany, pages 1525–1534.
- Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2011. English gigaword. *Linguistic Data Consortium* .
- Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. *NIPS 2017 Autodiff Workshop* .
- Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. volume 14, pages 1532–1543.
- Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *NAACL*. volume 1, pages 2227–2237.

- Hieu Pham, Minh-Thang Luong, and Christopher D. Manning. 2015a. Learning distributed representations for multilingual text sequences. In *Workshop on Vector Space Modeling for NLP*.
- Nghia The Pham, Germán Kruszewski, Angeliki Lazaridou, and Marco Baroni. 2015b. Jointly optimizing word representations for lexical and sentential tasks with the C-PHRASE model. In *Proceedings of ACL*. Beijing, China, pages 971–981.
- Lison Pierre and Tiedemann Jörg. 2016. Pierre lison and jörg tiedemann, 2016, opensubtitles2016: Extracting large parallel corpora from movie and tv subtitles. In *LREC*.
- Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. In *NAACL*.
- N. Pourdamghani and K. Knight. 2017. Deciphering related languages. In *EMNLP*.
- Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.
- Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners .
- Miloš Radovanović, Alexandros Nanopoulos, and Mirjana Ivanović. 2010. Hubs in space: Popular nearest neighbors in high-dimensional data. *Journal of Machine Learning Research* 11(Sep):2487–2531.
- Prajit Ramachandran, Peter J Liu, and Quoc V Le. 2016. Unsupervised pretraining for sequence to sequence learning. *arXiv preprint arXiv:1611.02683* .
- Marc’Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2015. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732* .
- Reinhard Rapp. 1995. Identifying word translations in non-parallel texts. In *Proceedings of the 33rd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, ACL ’95, pages 320–322.
- Reinhard Rapp. 1999. Automatic identification of word translations from unrelated english and german corpora. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, ACL ’99.

- S. Ravi and K. Knight. 2011. Deciphering foreign language. In *ACL*.
- Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. 2014. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. pages 806–813.
- Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiský, and Phil Blunsom. 2016. Reasoning about entailment with neural attention. *ICLR*.
- Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. 2000. The earth mover’s distance as a metric for image retrieval. *International journal of computer vision* 40(2):99–121.
- Rafael Sabatini. 1922. *Captain Blood*. Houghton Mifflin Company.
- Charles Schafer and David Yarowsky. 2002. Inducing translation lexicons via diverse similarity measures and bridge languages. In *Proceedings of the 6th Conference on Natural Language Learning - Volume 20*. Association for Computational Linguistics, COLING-02.
- Peter H Schönemann. 1966. A generalized solution of the orthogonal procrustes problem. *Psychometrika* 31(1):1–10.
- Holger Schwenk and Xian Li. 2018. A corpus for multilingual document classification in eight languages. In *LREC*. pages 3548–3551.
- Holger Schwenk, Ke Tran, Orhan Firat, and Matthijs Douze. 2017. Learning joint multilingual sentence representations with neural machine translation. In *ACL workshop, Repl4NLP*.
- Rico Sennrich. 2017. How grammatical is character-level neural machine translation? assessing MT quality with contrastive translation pairs. In *Proceedings of EACL (Short Papers)*. Valencia, Spain, pages 376–382.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015a. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. pages 1715–1725.

- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Edinburgh neural machine translation systems for wmt 16. *arXiv preprint arXiv:1606.02891* .
- Xing Shi, Inkit Padhi, and Kevin Knight. 2016. Does string-based neural MT learn source syntax? In *Proceedings of EMNLP*. Austin, Texas, pages 1526–1534.
- Laura Smith, Salvatore Giorgi, Rishi Solanki, Johannes C. Eichstaedt, H. Andrew Schwartz, Muhammad Abdul-Mageed, Anneke Buffone, and Lyle H. Ungar. 2016. Does 'well-being' translate on twitter? In *EMNLP*. pages 2042–2047.
- Samuel L Smith, David HP Turban, Steven Hamblin, and Nils Y Hammerla. 2017. Offline bilingual word vectors, orthogonal transformations and the inverted softmax. *International Conference on Learning Representations* .
- Richard Socher, Eric Huang, Jeffrey Pennin, Andrew Ng, and Christopher Manning. 2011. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *Proceedings of NIPS*. Granada, Spain, pages 801–809.
- Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*. pages 1631–1642.
- Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research* 15(1):1929–1958.
- Sandeep Subramanian, Adam Tsirischler, Yoshua Bengio, and Christopher J Pal. 2018. Learning general purpose distributed sentence representations via large scale multi-task learning. In *International Conference on Learning Representations*.
- Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. End-to-end memory networks. In *Advances in neural information processing systems*. pages 2440–2448.
- Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*. pages 3104–3112.

- Kai Sheng Tai, Richard Socher, and Christopher D Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Yaniv Taigman, Ming Yang, Marc’Aurelio Ranzato, and Lior Wolf. 2014. Deepface: Closing the gap to human-level performance in face verification. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, page 8.
- Shuai Tang, Hailin Jin, Chen Fang, Zhaowen Wang, and Virginia R de Sa. 2017. Trimming and improving skip-thought vectors. *Proceedings of the 2nd Workshop on Representation Learning for NLP*.
- Wilson L Taylor. 1953. “cloze procedure”: A new tool for measuring readability. *Journalism Bulletin* 30(4):415–433.
- Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC’12)*. European Language Resources Association (ELRA), Istanbul, Turkey.
- Masatoshi Tsuchiya. 2018. Performance impact caused by hidden bias of training data for recognizing textual entailment. In *LREC*.
- Peter D Turney. 2012. Domain and function: A dual-space model of semantic relations and compositions. *Journal of Artificial Intelligence Research* 44:533–585.
- Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. 2017. Deep image prior. <https://arxiv.org/abs/1711.10925>.
- Shinji Umeyama. 1988. An eigendecomposition approach to weighted graph matching problems. *IEEE transactions on pattern analysis and machine intelligence* 10(5):695–703.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*. pages 6000–6010.

- Ivan Vendrov, Ryan Kiros, Sanja Fidler, and Raquel Urtasun. 2016. Order-embeddings of images and language. *International Conference on Learning Representations* .
- Oriol Vinyals, Łukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. 2015a. Grammar as a foreign language. In *Advances in Neural Information Processing Systems*. pages 2773–2781.
- Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015b. Show and tell: a neural image caption generator. In *Proceedings of CVPR*. Boston, MA, pages 3156–3164.
- Ellen M Voorhees and Dawn M Tice. 2000. Building a question answering test collection. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, pages 200–207.
- Ivan Vulić, Douwe Kiela, and Anna Korhonen. 2017. Evaluation by association: A systematic study of quantitative word association evaluation. In *Proceedings of EACL*. volume 1, pages 163–175.
- Ivan Vulic and Marie-Francine Moens. 2015. Bilingual word embeddings from non-parallel document-aligned data applied to bilingual lexicon induction. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL 2015)* pages 719–725.
- Takashi Wada and Tomoharu Iwata. 2018. Unsupervised cross-lingual word embedding by multi-lingual neural language models. *arXiv preprint arXiv:1809.02306* .
- Alexander Waibel, Toshiyuki Hanazawa, Geoffrey Hinton, Kiyohiro Shikano, and Kevin J Lang. 1995. Phoneme recognition using time-delay neural networks. *Backpropagation: Theory, Architectures and Applications* pages 35–61.
- Alex Wang, Amapreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461* .
- Paul J Werbos. 1990. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE* 78(10):1550–1560.
- Jason Weston, Samy Bengio, and Nicolas Usunier. 2011. Wsabie: Scaling up to large vocabulary image annotation. In *IJCAI*.

- Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language resources and evaluation* 39(2):165–210.
- John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2015. Towards universal paraphrastic sentence embeddings. *Proceedings of the 4th International Conference on Learning Representations (ICLR)* .
- John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2016. Charagram: Embedding words and sentences via character n-grams. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)* .
- Adina Williams, Nikita Nangia, and Samuel R Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of NAACL*.
- Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc 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. 2016. Google’s neural machine translation system: Bridging the gap between human and machine translation. <http://arxiv.org/abs/1609.08144>.
- Min Xiao and Yuhong Guo. 2014. Distributed word representation learning for cross-lingual dependency parsing. In *CoNLL*. pages 119–129.
- Chao Xing, Dong Wang, Chao Liu, and Yiye Lin. 2015. Normalized word embedding and orthogonal transform for bilingual word translation. *Proceedings of NAACL* .
- Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of ICML*. Lille, France, pages 2048–2057.
- Lihi Zelnik-manor and Pietro Perona. 2005. Self-tuning spectral clustering. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, MIT Press, pages 1601–1608.

- Meng Zhang, Yang Liu, Huanbo Luan, and Maosong Sun. 2017a. Adversarial training for unsupervised bilingual lexicon induction. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics* .
- Meng Zhang, Yang Liu, Huanbo Luan, and Maosong Sun. 2017b. Earth mover’s distance minimization for unsupervised bilingual lexicon induction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 1924–1935.
- Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*. pages 649–657.
- Yuan Zhang, David Gaddy, Regina Barzilay, and Tommi Jaakkola. 2016. Ten pairs to tag—multilingual pos tagging via coarse mapping between embeddings. In *NAACL*. pages 1307–1317.
- Han Zhao, Zhengdong Lu, and Pascal Poupart. 2015. Self-adaptive hierarchical sentence model. In *Proceedings of the 24th International Conference on Artificial Intelligence, AAI Press, IJ-CAI’15*, pages 4069–4076.
- Jie Zhou, Ying Cao, Xuguang Wang, Peng Li, and Wei Xu. 2016a. Deep recurrent models with fast-forward connections for neural machine translation. *arXiv preprint arXiv:1606.04199* .
- Peng Zhou, Zhenyu Qi, Suncong Zheng, Jiaming Xu, Hongyun Bao, and Bo Xu. 2016b. Text classification improved by integrating bidirectional lstm with two-dimensional max pooling. *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics* .
- Xinjie Zhou, Xiaojun Wan, and Jianguo Xiao. 2016c. Cross-lingual sentiment classification with bilingual document representation learning. In *ACL*.
- Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*. pages 19–27.
- Michal Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. 2016. The united nations parallel corpus v1. 0. In *LREC*.

Will Y Zou, Richard Socher, Daniel M Cer, and Christopher D Manning. 2013. Bilingual word embeddings for phrase-based machine translation. *Proceedings of EMNLP* .