data2vec : A General Framework for Self-Supervised Learning in Speech, Vision and Language: Final Report

Chloé Sekkat
ENSAE Paris & ENS Paris-Saclay
chloe.sekkat@ensae.fr

Mathilde Kaploun
ENSAE Paris
mathilde.kaploun@ensae.fr

Abstract

This project is part of the Deep Learning course at ENSAE Paris. The authors have chosen to work on the first multimodal self-supervised algorithm, namely, data2vec [1]. It has been developed by FAIR quite recently and has shown promising results in speech, NLP and computer vision. The goal of this project is two-folds: get a better understanding of some keys concepts and works done in the field of self-supervised learning and study the performance of data2vec on several tasks in NLP and compare it to the best single-purpose algorithms in each domain.
Self-supervised algorithms are able to learn directly from the unlabeled input data by proposing clever systems of learning. Recently they provoked a revolution in the machine learning/deep learning world by enabling training on vast amount of unlabeled data and models trained in this fashion have beaten previous state of the art (SOTA) methods. This is the case, for instance, for the well-known BERT [10], a bidirectional encoder using a masked language modeling task and jointly conditioning on both left and right context in all layers. As the need to build smarter models able to be more generalist and make use of the underlying structure of the data is crucial, we decided to focus on self-supervised learning for this final project.

1. Problem Definition

Self-supervised learning is motivated by the goal of making use of large amount of unlabeled data and uncover their underlying structure. It is inspired by the way humans learn, namely, observation. A large part of what we learn as human come from us observing the world around us; not only the visual world but also sounds and language. We are constantly evolving thanks to our observations: there are key concepts in each objects/word/sound that we are able to grasp, experience on and then act on. It is our perception of the world and the associated common sense that allow humans to learn these basic concepts. These concepts are the building blocks of how humans learn and help us learn faster (think of a baby learning what "mommy" or "daddy" means).
On the opposite side, deep learning algorithms need a lot of examples and hours of training before being able to associate a phoneme to its written form. The goal of self-supervised learning is to design algorithms able to mimic this human common sense. A lot of research has been done in this field over the last 5 years, given rise to models such as BERT [10] in NLP or BeiT [2] in Computer Vision; models building on Transformers and self-attention [30].

However one challenge of self-supervised learning is that all developed methods are unimodal. This means that each training scheme/design is task-specific. For instance, there is masked language modeling for NLP and learning representations invariant to data augmentation in Computer Vision. Consequently, by the choice of these designs, biases are incorporated in each model and there is no unified way of learning. However if we think about how humans learn, we use similar processes for everything. This is one of the core motivation of Baevski et al. [1]. They would like to design an algorithm that would not be modality-specific i.e. that would benefit from cross-modal representations. They present data2vec as an algorithm that can learn contextualized latent representations instead of modality-specific representations. Concretely, the learning objective in DATA2VEC is the same across all modalities: masked learning to produce latent target *continuous and contextualized representations*, using a teacher-student architecture training scheme.

This project is organized as follows: Section 2 goes over important works done in the field of self-supervised learning. In Section 3 we shall describe how data2vec works and how it is different from what has been done before. The two next sections are dedicated to hands-on experiences on two NLP tasks: Question Answering (Section 4) and Sentiment Analysis (Section 5). The associated code is available on GitHub [1].

2. Literature Review

2.1. Self-supervised learning in general

As mentioned above in Section 1, self-supervised learning aims at learning the underlying representations of the input data to mimic the human common sense. The goal is to obtain a signal from the data by leveraging its underlying structure.

[1] https://github.com/chloeskt/dl_ensae

It makes use of large amounts of unlabeled data, hence alleviates the need for costly labelization. The resulting models are meant to be pre-trained using a particular task (masked language modeling for instance) and then finetuned on a particular downstream task using few labeled data (Question Answering for instance).

In the pre-training stage, the model will learn to predict or regress part of the input data using another part of it, usually masked. For instance, in masked language modeling, the model gets a sentence where a given percentage of the tokens/bytes/sub-words are masked and has to predict them (see [10]). This design requires only unlabeled data and converts easily an unsupervised learning problem (learning a representation of the input) into a supervised one (learning to predict words). By doing so, the model is expected to build an understanding of what is a word and how can it be placed/used in a given sentence (context).

Therefore self-supervised learning allows to reduce labeling cost and is a first step towards a generic artificial intelligence.

2.2. In Computer Vision

In computer vision, self-supervised learning is often used and developed in its contrastive form i.e. using both positive and negative examples. One core example of a simple contrastive loss is the triplet loss where given a triplet of two positives and one negative examples, one wants the model to have similar representations in the latent space of the two positives examples and that the representation of the negative one gets pulled apart. For a given image, the positive pair consists of two crops of this image and the negative example is a crop of a different image. One of the first paper to introduce such a contrastive method in an unsupervised framework was [24] with Contrastive Predictive Coding (CPC), developed by DeepMind. Combining an autoregressive model with probabilistic contrastive loss and negative sampling, CPC learns useful representations by trying to predict future samples. It was first designed for speech but can be extended and applied in other domains such as vision and language. For instance, in vision, the image is divided into a coarse grid and the model has to predict the lower rows of the same image using only the first rows (predict the "future"). When it was developed CPC outperformed SOTA unsupervised models in ImageNet Top-1% accuracy (48.7%), proving the quality of the latent representations it had learned.

A great improvement was proposed by Boostrap Your Own Latent (BYOL [13]), it allowed to reduce the gap between supervised and self-supervised methods. One of the key differences is that it gets rid of the negative samples. Instead, it uses two networks (can be seen as a siamese network) in student/teacher mode. For a given input image, two different views are generated (data augmentation); one is passed to the online model, the other to the target one. The two generated representations are then compared, the network minimizes a similarity loss between the two. The weights of the target network are an exponential moving average of those of the online network, allowing to retain only the most important features in the representation. One might note that the authors of BYOL [13] explain that there exists a collapsed solution for the objective but that the model avoids it (no given reason). Later, in the SimSiam paper [5], the authors show that this is due to the "stop-gradient" step ensuring that the target network never updates its weights through the computed gradients, preventing it to reach the collapsed point.

Transformers are also used in vision for self-supervised learning. Introduced by [12], Vision Transformers have a masked prediction objective, similar to what is done in NLP. Other works builds on these training objectives (BeiT [2]) and have outperformed SOTA supervised methods on ImageNet-1K. In ViT, each input image gets split into fixed-size patches which are then linearly embedded and are fed to a Transformer encoder jointly with patch and position embeddings. The authors note that compared to models based on convolution layers, ViT are less biased towards texture, hinting that their features contain more object-aware representations. Overall they are said to have less image-specific inductive bias and set, yet again, new SOTA scores.

Finally, one of the most interesting model, combining all ideas exposed above is DINO [4]. DINO builds on the self-distillation idea. It is a student/teacher training framework where, as in BYOL, the teacher's weights are an exponential average of those to the student. The student gets as input crops of the original image while the teacher gets the global views, this encourages "local-to-global" correspondences by enforcing the student to interpolate from a small context crop. A simple cross-entropy loss is used to compare the two output distributions and make them similar. The idea is to ask the student to have the same proportions of features as the teacher (the final softmax activation function can be seen as a way to convert raw activations to represent how much each feature is present relative to the whole). The authors in [4] show activation maps which illustrate a higher level image understanding compared to previous self-supervised models. They appear to contain explicit information about the semantic segmentation of objects in an image.

The key difference highlighted by data2vec authors [1] is that the representations provided by data2vec are truly contextualized. The latent target representations are based on the entire output while DINO, for instance, mostly focuses on learning transformation-invariant representations of pairs of visual tokens instead of structural information within the entire given sample.

2.3. In Natural Language Processing

Self-supervised learning has been used in Natural Language Processing (NLP) for quite some time, especially under the form of Language Models. One might say that self-supervised learning was popularized by the well-known Word2Vec article by Mikolov et al. [22] back in 2013. The article proposes two architectures for the Word2Vec algorithm:

the Continuous Bag of Words approach and the Skip-Gram one. The former takes as input a sequence of words of a chosen window size. Each center word gets masked so that the algorithm (a simple Multi-Layer Perceptron) has to predict it using the surrondings words (context). The Skip-Gram approach differs in that given a center word, the algorithm has to predict the surrounding words.
Next, these predicting task extended to sentences as in [16] for instance (neighbor sentence prediction). Given a center sentence, the algorithm has to predict the previous and the next sentences.

In the previous approaches, we make uses of both right and left context. However many self-supervised pre-training tasks actually only use left-to-right context. This is the case of auto-regressive language modeling which has been used in many papers (among those are GPT-1 and GPT-2 [25]). The great advantage of models trained of this task is that they can be used easily for text generation as they can be run sequentially. However a limit often pointed out with these type of auto-regressive language models is that one cannot make use of both right and left context simultaneously. This can be problematic as linguists have stressed out that the way humans construct sentences is based on their knowledge of the whole idea they want to express in this sentence and hence a given word is impacted both by the previous and the next words.

With that in mind came Attention [30]. This mechanism allows to look at the input sentence and decide, at each step, which parts of it are relevant for the task at hand. Transformers and especially Bi-directional ones [10] have become standard and SOTA in many tasks. Often they are trained using Masked Language Modeling ([10], [20], [8], [18]). A certain percentage of the tokens get masked, randomly, and the model has to predict them using left and right parts of the input sequence. One of the biggest difference with the previous approach is that here, the set of words to predict is actually finite. Only the masked words get to be predicted and only an understanding of the current sequence is needed (or at least the notion of sentence relations is not key for this task). Which is why BERT added a second objective in their pre-training: Next Sentence Prediction. The latter can be seen as a form of contrastive learning: given 3 sentences, 2 being consecutive and 1 from another document, the network has to decide whether or not the sentences can come one after another.

ALBERT [18] proposes another pre-training task instead of Next Sentence Prediction: Sentence Order Prediction. This variation takes pairs of two sentences from the same document as input and creates variations where the order of the sentences are swaped. The network has then to predict if the two sentences are in the correct order or not. Other pretraining tasks also focuses on sentence ordering and document structure in order for the model to build intuition on how sentences are related to one another and how documents are structured (Sentence Permutation, Document Rotation [19], [32]).

Similar models and training schemes have been used for cross-lingual tasks. For many of these models (mBERT, XLM [17], XLM-RoBERTa [8]), the underlying goal is to capture valuable linguistic information which could improve natural language understanding on text in a low-resource language or a language with different linguistic characteristics (richer morphology e.g.). XLM has proven to be very effective at capturing cross-lingual information and exploiting them in a vast range of downstream tasks. Often these multi-lingual models have proven to be better than regular unilingual models for a given task. This hints that cross-lingual sentence representations allow to be a step closer to a generic AI. Inside these representations must be a more general understanding of what the sentence is actually about, irrespective of the language it is in.

A downside of all the models mentioned above however is their size and inference time. A prime example is XLM-RoBERTa which is trained on 2.5TB of data from the internet. With this in mind, researchers have proposed works on knowledge distillation or self-knowledge distillation (DistilBERT [27]). This framework is meant to create a smaller model (student) which will capture and distil the knowledge of a bigger model (the teacher) without a too big loss in performance (model compression). Again, we face the same teacher/student training scheme exposed in the previous subsection. There are various types of knowledge distillation. First one needs to define "knowledge". The latter can be response-based (the student focuses on the last layer of the teacher), feature-based knowledge (focus on intermediate layers) or relation-based (focuses on relationship between activation maps). Finally there are three types of training scheme: offline distillation (teacher already pre-trained, only student has to be trained), online distillation (both have to be trained) and self-distillation (the same model is used as teacher and student). Authors of DistilBERT [27] use a triplet loss combining a language modeling loss, a distillation loss and a cosine-distance one. Their model was able to retain 95% of BERT performance of GLUE benchmark [31] while having 40% less parameters than BERT-base and being 60% faster.

data2vec builds on these ideas and propose a refinement where not words/token/bytes but latent representations are predicted by a student based on an averaged of K top layers of the teacher. The targets are not predefined therefore their number if unbounded. This feature gives more expressive power to the model because it can easily adapt to all input examples (no problem of out-of-vocabulary words for instance). And again, the latent target representations are contextualized, which is not the case in BERT-like models.

3. DATA2VEC

3.1. General idea

Overall the idea behind data2vec is straight-forward but very powerful. The authors propose a training scheme enabling the creation of the first multi-modal self-supervised algorithm. It builds on the self-distillation framework which we mentioned while presenting DINO [4] and uses a standard

Transformer architecture. What truly differentiate data2vec from previous works is that it predicts contextualized and continuous latent representations and not specific targets such as token/subwords/bytes (NLP) or visual tokens (Computer Vision). The masked prediction task is done on latent target representations (which are thus not related to a single modality) and the model is train to predict multiple layers (not only the last one).

In student mode, the model encodes a masked version of the input which it will then have to use to predict the full input representation created in teacher mode. As in DINO and BYOL, teacher's weights are an exponentially decaying average of those of the student.

Note that while the targets are multi-modal, the inputs are not. Indeed, the authors use modality-specific feature encoders and specific masking strategies. This is a limit to the generic nature of data2vec.

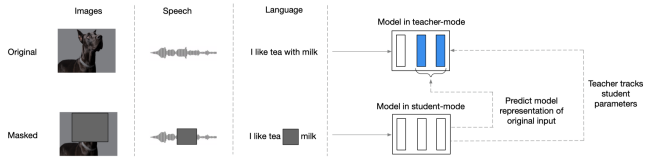

Figure 1. Illustration of how data2vec follows the same learning process for different modalities. The model first produces representations of the original input example (teacher mode) which are then regressed by the same model based on a masked version of the input. The teacher parameters are an exponentially moving average of the student weights. The student predicts the average of K network layers of the teacher (shaded in blue).

Figure 1. data2vec training scheme; extracted from the paper

3.2. Architecture

In each modality, the authors use a Transformer encoder. For instance, in vision they use a ViT [12] and the associated encoding steps. For NLP, they build on [10] using byte-pair encoding [28].

3.3. Processing & Masking

As mentioned above, the processing and masking steps are modality-specific.

In computer vision, the authors use the same encoding steps as ViT [12] (patches of 16x16) and the blockwise masking strategy described in BEiT [2]. The main difference is that they mask 60% of the patches instead of 40%, apply random data augmentations and feed both the student and the teacher the same modified image.

In NLP, the authors build on RoBERTa [20] and applied the same masking strategy as in BERT [10] to 15% of the tokens uniformly selected (tokens are created via BPE [28]).

3.4. Targets to learn

Targets are encoded over the whole input data using a Transformer. The self-attention ensures that the representation are contextualized and continuous. They are made of the last K blocks in the teacher network for time-steps which are masked in the student network. Each activation map extracted is normalized before averaging them all. The normalization step is said, by the authors, to prevent the model from collapsing (mode collapse) and to prevent layers with high norm to dominate the target features.

The student has to regress these targets through a smooth L1 loss. Given targets y_t, the objective writes:

$$\mathcal{L}\left(y_t, f_t(x)\right) = \begin{cases} \frac{1}{2}\left(y_t - f_t(x)\right)^2 / \beta & |y_t - f_t(x)| \leq \beta \\ \left(|y_t - f_t(x)| - \frac{1}{2}\beta\right) & \text{otherwise} \end{cases}$$

4. Question Answering

In this section, the goal is to evaluate the quality of the latent representations produced by data2vec against several models that have proven to be quite relevant for Question Answering task. The embeddings produced by data2vec are compared against both token-based models such as BERT [10], RoBERTa [20], mBERT, XLM-RoBERTa [8] and DistilBERT [27] and token-free models (character-based) such as CANINE [7]. Note that CANINE is a pre-trained tokenization and vocabulary-free encoder, that operates directly on character sequences without explicit tokenization. It seeks to generalize beyond the orthographic forms encountered during pre-training. data2vec is an algorithm that can learn a contextualized latent representations instead of modality-specific representations. Its learning objective is the same across all modalities: masked learning to produce latent target continuous and contextualized representations, using a teacher-student architecture training scheme.

We evaluate its capacities on extractive Question Answering (select minimal span answer within a context) on SQuADv2 dataset [26]. The latter is a unilingual (English) dataset. The two main metrics used are the F1 score and the Exact Match (EM) score. The obtained F1-scores are being compared to BERT-like models (BERT, DistilBERT, XLM-RoBERTa and mBERT) and CANINE. Note that mBERT, XLM-RoBERTa and CANINE were pre-trained on multilingual data. **Therefore data2vec is only directly comparable to BERT and RoBERTa, but it is still interesting to get the performances on other models on the same task.**

A second step of our analysis is to assess data2vec abilities to handle noisy inputs, especially noisy questions. The robustness to noise of such systems is imperative. It is highly probable that in real life settings, the Automatic Speech Recognition system (ASR) or the human typing the question actually do not produce qualitative text in the sense that the written-translation might be flawed (typos, misspellings, grammatical errors, etc).

Our third experiment consists in measuring the abilities of data2vec to adapt to new target domain by only doing few-shot learning. This means that we want to take a fine-tuned data2vec model (on SQuADv2 which is a general wikipedia-based dataset) and measure its performance on another domain-specific dataset (for instance medical or legal datasets which are two domains with very specific wording and concepts) after having train it for a small number of

	data2vec	RoBERTa	BERT	DistilBERT	mBERT	XLM-ROBERTA	CANINE-c	CANINE-s
Batch size	12	12	8	8	8	8	4	4
Learning Rate	2e-5	2e-5	3e-5	3e-5	2e-5	3e-5	5e-5	5e-5
Weigh decay	1e-4	1e-4	0	1e-2	0	0	0.01	0.001
Nb of epochs	3	3	2	2	2	2	3	2.5
Number of training examples	131823	131823	131754	131754	132335	133317	130303	130303
Number of validation examples	12165	12165	12134	12134	12245	12360	11861	11861
Max sequence length	348	348	348	348	348	348	2048	2048
Doc stride	128	128	128	128	128	128	512	512
Max answer length	30	30	30	30	30	30	256	256
Lr scheduler	cosine	cosine	none	linear	none	none	linear	linear
Warmup ratio	0.1	0.1	0	0.1	0	0	0.1	0.1

Table 1. Extensive list of hyperparameters used when finetuning models on SQuADv2 dataset. Note that models were trained on one Tesla P100 (16GB). Number of examples is given after the dataset has been tokenized.

epochs (3 or less) on a very small number of labeled data (less than 250 for instance). These performances will be compared to those of the other models we have chosen along this study.

Last, we will stay again in the few-shot learning domain but test the abilities of data2vec to resist to adversarial attacks knowing that it has not been trained for that and that it will only be trained for few epochs and a small number of adversarial examples.

4.1. Extractive Question Answering on SQuADv2

Extractive Question Answering consists in selecting the minimal span answer within a context to answer a given question. One of the most used datasets to benchmark models for this task is the SQuADv2 dataset [26] (Stanford Question Answering dataset). It is composed of 130 319 training examples and 11 873 validation examples. Not all questions have an answer in the associated context. Hence the model has to either predict the start and end character corresponding to the minimal answer or predict the equivalent of an empty string.

data2vec gives us final embeddings that we can feed to a single Linear layer of shape (768,2) to predict start and end logits for minimal span answer. Adding a final linear layer is done for every other model in this task. We also use the same structure. There is no dropout nor activation functions added (hence we retrieve logits). The loss is a simple cross-entropy. We make use of the Trainer class provided by HuggingFace [2] to train our models and made a wrapper around it to add more features. The code can be found on GitHub where a Command Line Interface has been exposed to reproduce our experiences.

Standard metrics to compare models on extractive Question Answering are F1-score and Exact Match (EM). The former measures the average overlap between the prediction and ground truth answer while the latter measures the percentage of predictions that match any one of the ground truth answers exactly. Hence the F1 score is always greater than the EM score.

To ensure fair comparison, we finetuned ourselves data2vec, BERT, DistilBERT, RoBERTa, CANINE-C (autoregressive character loss), CANINE-S (subword loss), mBERT and XLM-RoBERTa using pretrained models from HuggingFace (respectively FACEBOOK/DATA2VEC-TEXT-BASE, BERT-BASE-UNCASED, DISTILBERT-BASE-UNCASED, ROBERTA-BASE, GOOGLE/CANINE-C, GOOGLE/CANINE-S, BERT-BASE-MULTILINGUAL-CASED, XLM-ROBERTA-BASE). All models were trained using AdamW optimizer (improved weight decay compared to the classic Adam) and early stopping on validation set. All hyperparameters are given in Table 1. Table 2 summarizes the best F1-score we were able to get (on manually created test set).

	F1-score	EM score
data2vec	**78.21**	**74.90**
BERT	76.74	73.59
RoBERTa	**82.02**	**78.54**
DistilBERT	67.81	64.71
CANINE-C	74.1	69.2
CANINE-S	72.5	69.6
mBERT	77.51	74.1
XLM-RoBERTa	78.3	75.12

Table 2. F1 and Exact Match scores obtained on SQuADv2 manually created test sets. data2vec is comparable to XLM-RoBERTa and outperforms BERT. However RoBERTa has +3.81F1. Results are averaged over three runs.

data2vec performs decently well, reaching a F1 score of 78.21, right after RoBERTa (82.02). The former outperforms every other models by at least one point. What is noticeable is that multi-lingual token-based models perform better than unilingual ones on English dataset. This hints that cross-lingual features can help to get more accurate learning representation of words. data2vec performs even better than that. It might be due to the contextualized embeddings which contains higher level information. RoBERTa is even better. Our intuition is that this is due to the masking scheme applied, which differs from the one applied in BERT ([20]). In the former, the masking is done during training, the authors gen-

[2] https://huggingface.co/docs/transformers/main/en/main_classes/trainer#transformers.Trainer

erate masks on the sequence when it gets fed to the model. This means that the number of combinations of masked versions for a given sentence is way larger than the fixed number in BERT masking strategy.

4.2. Robustness to noise

The following experiment is motivated by the observation that when Question Answering systems are deployed, they face inherent noise which can heavily weight on their performances. For instance, Question Answering systems are often used in voice assistants. This implies that before the QA pipeline there is an ASR system which is, obviously, never perfect. Retranscribed questions have misspellings, grammatical errors, etc. Major types of noise can come from speech recognizers, keyboard typos and translation machines. In this experiment, we will address only misspellings-type of typos, but note that the code associated to this project allows for evaluation on other types of noise.

We created a noisy version of the SQuADv2 dataset where, given a noise level p, we transform each word into a noisy version thanks to the NLPAUG library [3]. We propose three levels of noise: 10%, 20% and 40%. We propose 4 types of noise: KEYBOARDAUG (mimics error due to the keyboard, substitutes character by keyboard distance), RANDOMCHARAUG (insert, swap, substitute or delete a character), OCRAUG (substitute a character by pre-defined OCR error) and BACKTRANSLATIONAUG (from English to German, then German to English).

For compute reasons, we only tested the RANDOMCHARAUG-substitution noise. The noise is applied on the SQuADv2 test set while the models are finetuned on a clean version of the training set (we took the finetuned models obtained in the previous subsection). Results are given in Table 3. Again RoBERTa seem to be the best model at hand. It is the most robust across the three noise levels. data2vec also appears to be quite robust compared to other models, it is often second or third in line. When $p = 20\%$, it is quite close to the performances of XLM-RoBERTa (while having less parameters). One might notice that as the level of noise increases, CANINE-S model actually reaches the second best score (close to data2vec, 67.18 vs. 67.14). This is quite interesting. It hints that tokenization-free models might be better to use in presence of noisy input (or out-of-vocabulary words since it is not bounded to a fixed vocabulary; the latter is a constraint of token-based models only). Moreover, this experience also highlights that the latent representations provided by data2vec are more robust to this keyboard-type of noise, regardless of the noise level. This makes it very competitive in real-life settings and also is a step closer to generic understanding and imitation of human capacities as we, humans, are able to understand the vast majority of a text even when it is noisy, especially when it is keyboard-type noise and random substitution of characters.

[3] https://github.com/makcedward/nlpaug

	Noise level 10%		Noise level 20%		Noise level 40%	
	F1 score	**EM**	**F1 score**	**EM**	**F1 score**	**EM**
data2vec	**75,1**	71,8	**72,53**	69,05	**67,14**	64,68
BERT	73,68	70,79	71,22	68,55	66,42	63,74
RoBERTa	**79,06**	75,87	**76,57**	73,56	**70,7**	68,18
DistilBERT	65,85	63,05	64,42	61,92	60,77	58,78
mBERT	74	70,75	71,66	68,46	67,08	64,74
XLM-RoBERTa	74,54	71,61	**72,68**	69,81	67,12	64,43
CANINE-C	69,64	66,89	67,88	65,43	66,03	63,9
CANINE-S	72,25	69,65	70,3	68,03	**67,18**	64,6

Table 3. F1 and EM scores reported on three noisy test sets extracted from SQuADv2. RoBERTa is the most robust model. Results are averaged over three runs.

4.3. Few-shot learning and domain adaptation

For this third experiment, we decided to test the models' abilities to rapidly switch to another domain than the one it was trained on. The idea is to apply few-shot learning on a dataset coming from another domain than SQuADv2. To do so, we have chosen the CONTRACT UNDERSTANDING ATTICUS DATASET (CUAD) [4] [14]. The goal of this experiment is to measure the ability of data2vec (and other models) to transfer to unseen data, in another domain. This could either be done in zero-shot or few-shot settings. Here we decided to go with the latter as it is more realistic. In real life, a company might already have a custom small database of labeled documents and questions associated (manually created) but would want to deploy a Question Answering system on the whole unlabeled database. The CUAD dataset is perfect for this task as it is highly specialized (legal domain, legal contract review). The training set is made of 22450 question/context pairs and the test set of 4182. We randomly selected 1% of the training set (224 examples) to train on for 3 epochs, using the previously finetuned models on SQuADv2. Then each model was evaluated on 656 test examples. Results are reported in Table 4 and to ensure fair comparison, all models where trained and tested on the exact same examples.

	F1 score	**EM score**
data2vec	**74.57**	72.24
BERT	74.18	72.72
RoBERTa	73.83	72.24
DistilBERT	72.86	71.37
mBERT	74.50	73.12
XLM-RoBERTa	**76.64**	73.44
CANINE-C	72.51	71.39
CANINE-S	72.27	71.27

Table 4. F1 scores and EM scores on 656 examples of CUAD's test set. Models have been trained for 3 epochs on 224 training examples. Results are averaged over three runs.

All models perform quite similarly except for XLM-RoBERTa which has + 2F1 compared to data2vec, second-best model in line. This experiment highlights the quality of

[4] https://github.com/TheAtticusProject/cuad

the latent representation produced by data2vec and their ability to quickly adapt to new domain. However, we were not able to see a great difference with the abilities of other BERT-like models.

4.4. Few-shot learning and adversarial attacks

This last Question Answering-related experiment aims at testing data2vec abilities not to be fooled in adversarial settings. We decided to us the dynabench/QA dataset [3] (BERT-version). The latter is an adversarially collected Reading Comprehension dataset spanning over multiple rounds of data collect. It has been made so that SOTA NLP models find it challenging. Dynabench collects human-in-the-loop data dynamically, against SOTA models such as BERT-Large for instance. Concretely, humans create questions adversarially, such that the model fails to answer them correctly. This work is inspired by several papers on Adversarial-QA ([23], [11], [34], [15]). Concretely the questions are hard in the sense that they do not derive directly from the context, the goal is to create a question that will "force" the model to create a real understanding of the context and of each word in it. A great example can be found in Figure 3 of paper [3].

In these papers, usually what is done is that models are trained both on SQuAD and an adversarial dataset, F1 score on test set should be improved as these adversarial dataset are made to increase generalization capabilities of the models by confronting them with hard/tricky questions. This is one of the results in [3]: models trained on adversarially collected samples show better generalization to non-adversarially collected samples. Here we decided to take models finetuned on SQuADv2, take 200 examples (2%) extracted from dynabench/qa training set to train each model for 3 epochs and then evaluate these models on 600 test examples (60% of the full test set). This is partly inspired by the results discussed in Table 7 of [3]. Our results are displayed in Table 6. Again, to ensure fair comparison, all models are trained on the exact same examples and evaluated on the same ones.

	F1 score	**EM score**
data2vec	**39.33**	29.6
BERT	38.13	25.6
RoBERTa	**47.47**	35.8
DistilBERT	32.64	22.5
mBERT	38.43	28.6
XLM-RoBERTa	36.51	27.6
CANINE-C	28.25	18.6
CANINE-S	27.40	17.2

Table 5. F1 scores and EM scores on 600 test examples from dynabench/qa dataset. Models have been trained for 3 epochs on 200 training examples. Results are averaged over three runs.

The results are comparable to those displayed in the mentioned Table 7. RoBERTa is the model with the best performance, with more than 8F1 points then the second-best in line, data2vec. Again, data2vec shows great abilities compared to BERT-like models but is never the first choice in terms of performance, either RoBERTa or XLM-RoBERTa model is often better. However, one might note that data2vec is smaller in terms of parameters hence faster to train (especially compared to XLM-RoBERTa which is known to be a huge model). This makes data2vec attractive.

Finally, we observed that CANINE models are much more prone to adversarial attacks (-10F1 points compared to data2vec and BERT). It is yet unclear for us why it is the case. Surely this is due to the fact that CANINE is tokenization-free but we still need to build intuition on why this has a great impact when evaluated on adversarial samples.

4.5. Discussion

This is part of the study we did a great number of experiments and research in order to test the abilities and limits of data2vec [1]. There are several teachings that can be remembered. First, as pointed out in the paper, data2vec is as powerful as BERT and even exceeds its capacities in all our experiments. However it still does not do better than RoBERTa on simple extractive Question Answering, noisy dataset (except for high level of noise) and adversarially collected dataset. In the domain adaptation experiment, XLM-RoBERTa is the best performing model. Often data2vec comes second which is why its capacities must not be downgraded. It is as the corresponding paper shows, virtually as good as BERT-like models while being the first model of its kind: a modal self-supervised model.

There are several experiments that one could also do in order to further test data2vec abilities. In the noise experiment for instance, we only added artificial noise. It could be interesting to add natural noise. Moreover, we only added noise to the test test. Another experiment could be to add noise to the training and the validation sets as well.

Another experiment could be to finetune data2vec (and other models) not only on SQuADv2 but also on the adversarial dynabench/qa dataset and assess whether or not the generalization of data2vec is improved (when tested on a non adversarial dataset).

Throughout our experiments we also noticed that often multilingual models are better than unilingual ones when evaluated in downstream tasks. This might because meaningful information can be derived from cross-lingual data where the model could build a representation of a word for its meaning rather than conditioned on the language it is written into. Therefore it could be interesting to see how a multi-lingual data2vec could perform (implying to pre-train it from scratch on, for instance, the top 104 Wikipedia-languages).

In the end, one should remember that these models are pretty heavy and memory hungry. This is a great limit to their use as one needs to have access to a great compute cluster.

5. Sentiment Analysis

5.1. Introduction

In this section, we tested the performance of Data2Vec on another classic NLP task : sentiment analysis. For this we chose to compare the performance of a fine-tuned classifier on two classical sentiment analysis datasets. First we chose the Imdb Movie Review Dataset first introduced by [21] in 2011 and which is the most used dataset for sentiment analysis benchmarking. It contains 50 000 rows of reviews, in English only, which are labellized as positive or negative. To check if our results were consistent we also fine-tuned Data2Vec on Airline reviews extracted from Twitter. The dataset was originally introduced by a company called Crowdflower.

We decided to test the performance on sentiment analysis against a classic embedder : Bert. It is not SOTA in terms of sentiment analysis, as Naive Bayes weighted bag of n-grams [29] and XL-Net [33] both perform better on the Imdb Reviews dataset. However, since its development by Google in 2018 [9] Bert has become a ubiquitous baseline in NLP experiments. Since 2020 it is also used by Google in most English-language queries for its search engine. Thus we thought it would provide a fair comparison for Data2Vec on sentiment analysis.

There are two versions of the original Bert : $\text{BERT}_{\text{BASE}}$ and $\text{BERT}_{\text{LARGE}}$. $\text{BERT}_{\text{BASE}}$ is constituted of 12 encoders with 12 bidirectional self-attention heads. $\text{BERT}_{\text{BASE}}$ is much bigger with 24 encoders with 16 bidirectional self-attention heads. For computational reasons, we chose to fine-tune $\text{BERT}_{\text{BASE}}$.

5.2. Architecture

In order to fine-tune our pre-trained models and adapt them to the classification task, we have created and hybrid architecture. The full models thus are composed of the existing architecture of either Data2Vec or Bert, to which we have added two Linear layers linked by a ReLu activation function. The first layer is our classifying layer and learns the link between the weights of the data once gone through the main architecture, and the label. The last layer serves no other purpose than to go from the weight space to our label space, which is a single label between 0 and 1 (0 representing a negative sentiment and 1 a positive one).

5.3. Training

For the fine-tuning, we have chosen to retrain the full architecture in both case. This significantly lengthens the training process, since all weights in the considerable Bert and Data2Vec architecture have to be optimized, instead of the two layers of our small classifier component. However, we thought it was especially interesting for Data2Vec as the architecture is not meant specifically for textual data, and thus we stand to gain more from fine-tuning.

As a loss, we chose to use Cross Entropy which is the most usual for classification tasks. For optimizer, we chose AdamW. It is a variant of the optimizer Adam with an improved implementation of weight decay. Weight decay is a form of regularization which lowers the chance of overfitting. As hyperparameters we chose 5e-5 for the learning rate, which performed well, and 1e-8 for weight decay.

We also added an early stopper with patience of 5 since the initial choice of 4 epochs was superfluous for all the models' training. This is probably due to the fact that our weights were already close to the optimal ones.

5.4. Results

We chose to evaluate the classification performances of each model using the Matthew Correlation Coefficient (or phi coefficient). It's robust to imbalance in class size. It can be interpreted as a Pearson's correlation coefficient between the observed and predicted binary classification. If we note TP for true positives, TN for True Negatives, FP for False Positives and FN for false negatives, we can compute it using a simple confusion matrix :

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

It is regarded as the most robust measure of binary classification model quality [6] as it only produces a high score if the prediction obtains good results in all confusion matrix categories. It is proportional to the size of the positive and negative elements which makes it also robust to category size. In our case we don't make use of this feature as our datasets are balanced.

	Imdb	**Airlines**
data2vec	0.70	0.78
BERT	0.77	0.86

Table 6. Matthew's Correlation Score on Imdb Review and Airline Complaints dataset . Models have been trained for 4 epochs on 10 000 training examples with early stopping.

We can see that Bert performed significantly better than Data2vec for both datasets. This could be surprising given that in question answering Data2Vec performed consistently better than Bert. It seems that for the simplicity of a classification task Bert's specialisation in language works at its advantage. We can also note that Bert is faster to train than Data2Vec. However, given Data2Vec's versality the results are still remarkable.

6. Acknowledgments

We would like to thanks Prof. Marco Cuturi for this Deep Learning course at ENSAE Paris and for letting us work on our chosen topic. It was a great opportunity for us to learn many things, both in the academic-research and practical (computer science) domains.

References

[1] A. Baevski, W.-N. Hsu, Q. Xu, A. Babu, J. Gu, and M. Auli. data2vec: A general framework for self-supervised learning in speech, vision and language, 2022.

[2] H. Bao, L. Dong, and F. Wei. Beit: Bert pre-training of image transformers, 2021.

[3] M. Bartolo, A. Roberts, J. Welbl, S. Riedel, and P. Stenetorp. Beat the AI: Investigating adversarial human annotation for reading comprehension. *Transactions of the Association for Computational Linguistics*, 8:662–678, dec 2020.

[4] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin. Emerging properties in self-supervised vision transformers, 2021.

[5] X. Chen and K. He. Exploring simple siamese representation learning, 2020.

[6] D. Chicco and G. Jurman. The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC Genomics*, 21(1):6, 2020.

[7] J. H. Clark, D. Garrette, I. Turc, and J. Wieting. CANINE: Pre-training an efficient tokenization-free encoder for language representation, 2021.

[8] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov. Unsupervised cross-lingual representation learning at scale, 2019.

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018.

[10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[11] E. Dinan, S. Humeau, B. Chintagunta, and J. Weston. Build it break it fix it for dialogue safety: Robustness from adversarial human attack. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4537–4546, Hong Kong, China, Nov. 2019. Association for Computational Linguistics.

[12] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

[13] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar, B. Piot, K. Kavukcuoglu, R. Munos, and M. Valko. Bootstrap your own latent: A new approach to self-supervised learning, 2020.

[14] D. Hendrycks, C. Burns, A. Chen, and S. Ball. Cuad: An expert-annotated nlp dataset for legal contract review, 2021.

[15] R. Jia and P. Liang. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark, Sept. 2017. Association for Computational Linguistics.

[16] R. Kiros, Y. Zhu, R. Salakhutdinov, R. S. Zemel, A. Torralba, R. Urtasun, and S. Fidler. Skip-thought vectors, 2015.

[17] G. Lample and A. Conneau. Cross-lingual language model pretraining, 2019.

[18] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut. Albert: A lite bert for self-supervised learning of language representations, 2019.

[19] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, 2019.

[20] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

[21] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[22] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space, 2013.

[23] Y. Nie, A. Williams, E. Dinan, M. Bansal, J. Weston, and D. Kiela. Adversarial nli: A new benchmark for natural language understanding, 2019.

[24] A. v. d. Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding, 2018.

[25] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners. 2019.

[26] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang. SQuAD: 100,000+ Questions for Machine Comprehension of Text. *arXiv e-prints*, page arXiv:1606.05250, 2016.

[27] V. Sanh, L. Debut, J. Chaumond, and T. Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[28] R. Sennrich, B. Haddow, and A. Birch. Neural machine translation of rare words with subword units, 2015.

[29] T. Thongtan and T. Phienthrakul. Sentiment classification using document embeddings trained with cosine similarity. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 407–414, Florence, Italy, July 2019. Association for Computational Linguistics.

[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need, 2017.

[31] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

[32] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le. Xlnet: Generalized autoregressive pretraining for language understanding, 2019.

[33] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le. Xlnet: Generalized autoregressive pretraining for language understanding, 2019.

[34] R. Zellers, Y. Bisk, R. Schwartz, and Y. Choi. SWAG: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 93–104, Brussels, Belgium, Oct.-Nov. 2018. Association for Computational Linguistics.