ITEA 3

PAPUD ITEA3 - 16037

Profiling and **A**nalysis **P**latform **U**sing **D**eep Learning

D1.3 State of The Art

Release	:	1.1
Status	:	Final
Date	:	6 May 2019
Author	:	Samin Mohammadi - Praboda Rajapaksha (IMT-TSP)
Visibility	:	Public
Pages	:	71

HISTORY OF CHANGES

Version	Date	Changes	Author
V 0.1	03/04/2018	First draft - ToC	Samin Mohammadi - Praboda Rajapaksha
V 0.2	13/04/2018	Finalized ToC	Samin Mohammadi
V 0.3	23/04/2018	Added Turk Telekom's SoTA	Engin Zeydan
V 0.4	25/04/2018	Added Prescriptive maintenance (ATOS) SoTA	Marc Platini
V 0.5	25/04/2018	Added LORIA contribution	Christophe Cerisara
V 0.6	25/04/2018	Press'Innov contribution	Samir Amir
V0.7	25/04/2018	IMT Contribution to Ch3	Samin Mohammadi
V0.8	26/04/2018	Merged initial contributions	Samin Mohammadi
V0.9	03/05/2018	Add contributions for BAREM related SoTA (Log file and sentiment)	Laurent Goncalves Marius Bilasco
V0.10	04/05/2018	DL distribution on many cores machines	Matthieu Ospici
V0.11	07/05/2017	IMT contribution to Chapter 3	Praboda Rajapaksha

V0.12	11/05/2017	BEIA contributions	George Suciu
V0.13	11/05/2017	Contribution to 4.3	-
V0.14	11/05/2017	Integrated contributions	Samin Mohammadi
V0.15	14/05/2017	HIB contribution to 3.4	Tamara Martín Elena Muelas
V0.16	17/05/2018	Added Kuleuven contribution	Quynh Do
V0.17	17/05/2018	Merged contributions	Samin mohammadi
V0.18	18/05/2018	Added BAREM references	José Mennesson
V0.19	18/05/2018	BEIA contributions	-
V0.20	24/05/2018	KU contribution to 3.2.3	Quynh Do
V0.21	24/05/2018	IMT- Added Introduction and Conclusion	Samin Mohammadi
V0.22	24/05/2018	LORIA - Added captions to figures in section 3.2.2.	Samuel Cruz-Lara
V0.23	25/05/2018	Univ. Lille - BAREM references	José Mennesson
V0.24	28/05/2018	Contribution to 3.2	Quynh Do
V0.25	31/05/2018	Added references	Samin Mohammadi
V0.26	07/06/2018	Added contribution to 4.1	Güneş Söyler
V0.27	08/06/2018	Merged version to review	Samin Mohammadi
V0.28	12/06/2018	Reviewed	Christophe Cerisara
V0.29	25/06/2018	Reviewed	Güneş Söyler
V1.0	02/07/2018	Final version	Samin Mohammadi
V1.1	06/05/2019	Final and public version	Cornel Crisan

ABSTRACT

This document surveys the state of the art on deep learning techniques and methods which will be mainly employed in Papud use cases including call centers operations, e-commerce, user behavior and human resources analysis, and prescriptive maintenance for HPC.

Table of Contents

List of Figures

1 Executive summary

Deep learning as a collection of techniques has had a great impact on various fields including Natural Language Processing (NLP), computer vision, audio and speech recognition, social networks analysis, and so on. The PAPUD project aims to exploit these advancements in deep learning methods in order to build universal models for data analytics. This deliverable has mainly focused on exploring the application of existing deep learning architectures, methods, and solutions in the following domains: media, finance, E-commerce, telecommunication, E-government, HR, and social media.

The main topics covered by this document include state-of-the-art in distributed deep learning on many core machines, where we discuss the challenges and solutions on data and model parallelism. Further, different deep learning frameworks are presented and performance evaluation methods are discussed.

From a more technical perspective, the document has a dedicated section to the state-of-the-art on deep learning which will present available methods and tools on data analytics using deep learning methods. Existing researches and studies related to big data analytics, natural language processing, text analytics, recommender systems, call centers, and online social networks (OSNs) analytics are explored and their applications in the PAPUD project are discussed.

As the PAPUD use cases comprise different concepts of deep learning, topics and studies related to each use case are introduced in different subsections as follows:

- Improve Customer experience in e-commerce
- Call center operations
- Human resources use case
- Behavior analysis for reserve efficient modeling
- Prescriptive maintenance for HPC

2 State of the art on deep learning distribution on many cores machines

Because training deep learning models require a lot of computing power, it can take advantage of cluster of computing resources (HPC cluster for example). In this section a computing resource (CR) means a GPU or a compute node equipped with several CPU cores.

To take advantage of a cluster, we need to distribute the computations across the computing resources. The first section details this problem.

To have the best performance, we need to exploit all the computing power of each computing resources. This is the purpose of the second section.

2.1 Distribution on Cluster

The training phase of deep learning can be distributed across several computing resources (GPU, nodes…). Two main approaches exist: model and data parallelism.

2.2 Data Parallelism

With data parallelism, the neural network is duplicated across all CR but each of them processes only a subset of the training data.

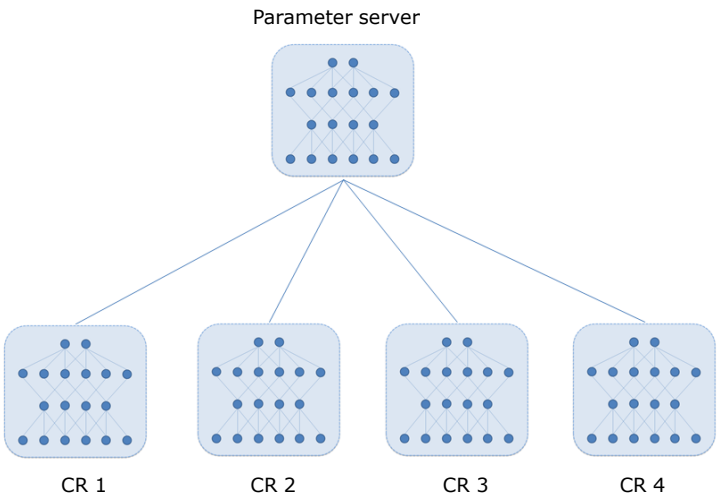

Figure 1 - Data parallelism

Data parallelism can be implemented in different ways. For example, Google has implemented a distributed SGD (called Downpour SGD) based on the structure shown in **Erreur ! Source du renvoi introuvable.** in the DistBelief project [1]. In this implementation, a parameter server asynchronously serves all of the computing resources.

Other implementations, without parameter server exist. For example, Baidu performs communication between all CR with a MPI_ALLREDUCE solution [2].

But, regardless of the implementation chosen, all parameters of the network must be regularly exchanged between computing resources. Therefore, large data transfers are mandatory (order of magnitude: several GB in medium/large neural network between the parameter server and CRs or between CRs).

2.3 Model Parallelism

With model parallelism (**Erreur ! Source du renvoi introuvable.**), the neural network is distributed across the computing resources. That means each computing resource contains one partition of the network. Conversely to data parallelism, data are exchanged between the neurons which are located at the boundary of each partition only.

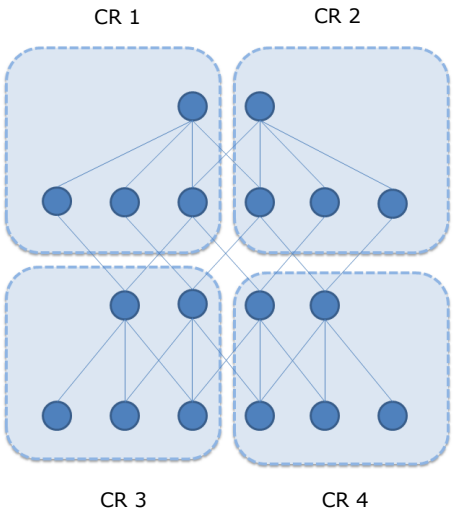

Figure 2 - Model parallelism

Model parallelism can be useful when we want to train a large neural network that does not fit on a single CR memory. It is difficult to predict the quantity of data exchange since it depends on the model topology. Nevertheless, with model parallelism the latency of the communication network will be very important since many communications occur.

2.3.1 Discussion

Data parallelism is easily implementable but can lead to very big data exchanges for large neural networks. Furthermore, data parallelism is not possible when a neural network does not fit on the memory of one CR.

With model parallelism, the amount of data exchanged can be considerably smaller compared to data parallelism. Nevertheless, models with local connectivity structure are more efficient compared to fully connected models given their lower communication requirements [1].

Some works try to combine model and data parallelism to increase the efficiency of parallelization [3].

These technics are generally used for training. Inference needs less computing power and can be done on other types of hardware. Nevertheless, with very large models, model parallelism should be done in inference phase since the memory of one CR cannot handle the entire model.

2.3.2 Parallelization on a compute node

An efficient way to distribute deep learning on several computing resources is essential but having good performance on each computing resource is also critical. In this area, Intel and NVIDIA develop hardware and software libraries designed for deep learning.

2.3.2.1 *GPU NVIDIA*

NVIDIA proposes hardware architecture optimized for deep learning, such as the DGX2 based on 8 Volta cards. NVIDIA also supports 16bits precision floating point for deep learning computations. The reduction of precision leads to frequent ambiguity solved by rounding up or down randomly (stochastic rounding) [4].

For inference, NVIDIA has released low power SOC (ARM + GPU) such as the Jetson TX1 (for embedded devices) or the DRIVE PX 2 (for autonomous cars).

NVIDIA develops a software library called cuDNN [5] designed to accelerate the computations performed in deep learning frameworks. Furthermore, since deep learning mainly uses linear algebra, many frameworks use NVIDIA's linear algebra libraries such as cuBLAS or cuSPARSE.

NVIDIA also develops a specific hardware interconnect called nvlink for inter-GPU high performance communication. The nccl library provides optimized primitives for collective multi-GPU communication.

2.3.2.2 *Intel*

Intel MKL and Intel DAAL full production release with CNN/DNN improvements are available. These libraries are optimized for multicore intel CPUs.

Furthermore, Intel works on interconnect and states that the Omnipath Interconnect will be efficient for distributed deep learning [6].

2.3.2.3 *Other companies*

Several companies work on processors designed for deep learning. For instance, Google has designed the TPU and TPU2 (Tensor Processing Unit), uses it in production and provides it as a service in its cloud services offer. Graphcore [1] company is developing the IPU (Intelligent Processing Unit) accelerator tailored for machine intelligence applications. Intel is developing a software stack name DLA (Deep Learning Accelerator) for accelerating the deep learning inference on its FPGA PAC card

[1] https://www.graphcore.ai/

(Programmable Accelerator Card). For the training, Intel will provide specific ASIC (Crest architecture) developed by its subsidiary Nervana.

2.4 Landscape of Deep Learning Frameworks

Many deep learning frameworks exist. We present here some of the most used frameworks.

2.4.1 Caffe

Caffe [7] is developed by Berkeley Vision and Learning Center (BVLC). It is a widely used machine-vision framework. Caffe is developed in C++ with efficient GPU support (in CUDA) but it has some binding for python or matlab. An experimental, community driven OpenCL version exists on the main Caffe source tree.

It defines a high level language (in protobuf) designed to easily implement deep neural network and data acquisition. One notable feature of caffe is a model repository called model zoo[2]. This repository contains hundreds of pre trained models which can be directly used for inference.

Caffe is mainly used for image processing (CNN) but some recent developments can help to process neural network for time series (RNN).

- http://caffe.berkeleyvision.org/
- https://github.com/BVLC/caffe

- Distribution / parallelization

Caffe can use multicore CPU with MKL or GPU with CUDA and cuDNN. Caffe supports multi-GPU on single node but it does not support distribution across computing node. Therefore, some companies have extended Caffe to support such distribution.

2.4.1.1 *CaffeOnSpark*

We can cite CaffeOnSpark[3], which was developed by Yahoo. It uses Spark to distribute the computations. To improve performances, CaffeOnSpark can directly access Infiniband cards for message exchange. CaffeOnSpark uses Data Parallelism for its distribution.

2.4.1.2 *Intel Caffe*

Intel Caffe is an optimized version of Caffe developed by Intel. Intel has optimized the CPU performance of Caffe by parallelizing the code with OpenMP and by optimizing the usage of MKL. A performance evaluation of Intel Caffe is available in section 2.5.2. Intel supports distribution (model and data parallelism) across computing node with MPI.

2.4.1.3 *Caffe2*

Caffe2 was started more recently and is developed by Facebook for running deep learning workload in production both in the datacenter and on mobile (cross platform library). It is natively distributed.

[2] https://github.com/BVLC/caffe/wiki/Model-Zoo
[3] https://github.com/yahoo/CaffeOnSpark

2.4.2 Torch /pytorch

Torch is used by companies like Facebook or Twitter. It is also mainly used by researchers on deep learning, because of its dynamic properties and flexibility to design novel models. Torch is a framework designed to make computations on tensor (n-dimensional matrix), with helper functions for deep learning. Some functions can, for example, load models from Caffe model zoo. Conversely to Caffe, the neural network needs to be written in a programming language called LUA. This choice leads to more flexibility but LUA suffers from a lack of adoption in the community. Pytorch has thus been proposed as a port of Torch to python. Both torch and pytorch are connected to the THNN[4] library for the computations

- Distribution / parallelization

Torch does not natively support distribution but some additional libraries, such as distlearn[5] exist. It uses the MKL for CPU parallelization and Cuda and cuDNN for GPU. It exploits multi-gpu on a single node.

- http://torch.ch/
- https://github.com/torch/torch7

2.4.3 Theano

Theano is one of the oldest deep learning framework. It has been developed by the MILA Lab at university of Montreal and it is suited for research on deep learning because of its flexibility. Numerous open-source deep-libraries have been built on top of Theano, including Keras [8]. Theano support has been abandoned in 2018.

- *Distribution* / parallelization

Theano uses the MKL for CPU parallelization and cuda and cuDNN for GPU. Theano does not support distribution across several compute nodes.

- http://deeplearning.net/software/theano/
- https://github.com/Theano/Theano

2.4.4 CNTK

CNTK is developed by Microsoft and works on Linux and Windows systems. As Caffe, CNTK implements a high level language to describe a neural network. It has been open-sourced in April 2015. CNTK can be used for different type of tasks such as image recognition, text processing… CNTK implements different types of neural network (CNN, RNN…).

- Distribution*/parallelization*

CNTK is designed to take advantage of multicore CPU and also single-GPU, multi-GPU, and multi-machine-multi-GPU. Furthermore, Microsoft has implemented a technic designed to reduce the communication cost [9].

The distribution is implemented with MPI.

[4] https://github.com/torch/nn/tree/master/lib/THNN
5 https://github.com/twitter/torch-distlearn

- http://www.cntk.ai/
- https://github.com/Microsoft/CNTK

2.4.5 TensorFlow

TensorFlow is developed by Google. It implements different types of neural networks like CNN or RNN. TensorFlow can be used with a Python or C++ API. A graphical interface, called TensorBoard can be used to easily view the training process and the different parameters of the neural network. Since 2017, a high-level interface, called Keras[6], is supported on the TensorFlow's core library.

- Distribution / parallelization

Like CNTK, tensorflow fully supports distribution on several computing nodes on CPUs or GPUs. The distribution solution has been released in March 2016 and is based on the GRPC[7] framework.

Some projects like horovod[8] aims to simplify the distributed training and relies on MPI for better performance.

2.4.6 MXNet

Apache MXNet is an open-source deep learning framework used to train, and deploy deep neural networks. It is scalable, allowing for fast model training, and supports a flexible programming model and multiple languages (C++, Python, JavaScript, Go, R, Scala, Perl)

- Distribution / parallelization

MXNET supports parallelization across cluster of compute nodes.

2.4.7 Discussion

In single node, all frameworks are usable and seem to be mature, but in distributed mode, it is more difficult to an easy- and ready-to-use solution. There are many bugs and lack of documentation since the distribution mode has been added only a few months ago in most framework.

For example, the implementation of the distribution on CaffeOnSpark or TensorFlow needs a lot of manual configuration and debugging to have a roughly working environment. The following Table is aimed to recap the information given on the lasts sections.

[6] https://www.tensorflow.org/api_docs/python/tf/keras
[7] http://www.grpc.io/
[8] https://github.com/uber/horovod

Table 1 - Deep learning frameworks characteristics

Framework	Core language	Bindings	GPU support	Distribution	Open Source	Pretrained Model	Caffe Parser	License	User base (+: small, ++: large, +++: very large)
Caffe	C++	Python, MatLab	x	Not native, with caffeonspark, intel Caffe	x	x		BSD	+++
Caffe2	C++	Python	X	X	X	X	X	Adhoc (Business friendly)	++
Torch	Lua		x	Not native	x		x	BSD	+++
Pytorch	Python	Python	x		x		x	BSD	+++
Theano	Python		x	not	x		limited to certain layers	BSD	++
CNTK	C++	Python (in development)	x	Yes with MPI	x			Microsoft	+
MXNet	C++	Python, C++, go …	x	x	x	x		Apache	+
TensorFlow	C++	Python	x	Yes with Grpc Yes with MPI (horovod)	X			Google	+++

2.5 Initial Performance Evaluation

ATOS made initial performance evaluation on Caffe, CaffeOnSpark and Intel Caffe. ATOS chooses to begin with Caffe because it's currently widely used on the industrial deep learning community. The performance evaluation is done on the training phase of the deep learning process.

2.5.1 Experimental conditions

2.5.1.1 *Hardware*

Our experimental hardware consists in a node with 2 GPUs NVIDIA K20X, 2 CPUs Intel E5-2695 v2 (2x12 cores) and 65GB of memory.

2.5.1.2 *Use case*

To evaluate the deep learning frameworks, we train a neural network designed to classify images across categories. To make our experiments we use a data set called "CIFAR10" [10]. It consists in 60000 RGB 32x32 pixel images, 50000 images for the training and 10000 to evaluate the trained model (test set).

The images are classified into 10 categories: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. Our neural network, trained with the CIFAR10 data set is a deep neural network with 4 layers. We use a special type of neural network called convolutional neural network [11]. These types of neural network give the best accuracy on image classification today.

2.5.2 CPU and GPU performance of Caffe

Firstly, we compare the CPU performance of Caffe against Intel Caffe. Then we compare the CPU performance against GPU performance.

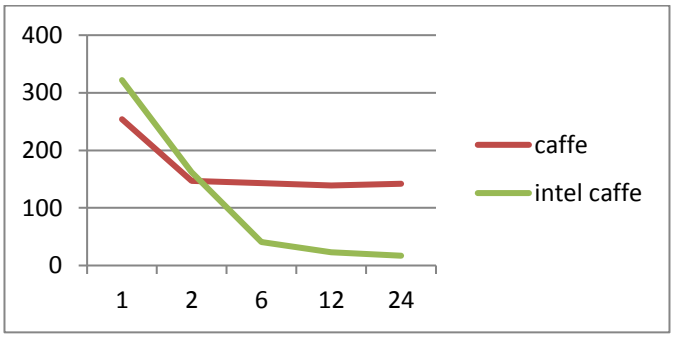

Figure 3 - Intel caffe and caffe on multicore CPUs

2.5.2.1 *CPU performance*

Erreur ! Source du renvoi introuvable. shows the time needed by Intel Caffe and Caffe to train the n etwork with 50000 images on our multicore CPU. As we can see, Intel Caffe shows the best performance: when the number of cores increases, Intel Caffe is able to take advantage of them. The

original Caffe implementation can't take advantage of more than 2 cores. Even if both Caffe and Intel Caffe are based on the MKL library for the CPU computations, the Intel implementation is considerably better. This means that it is not enough to use MKL to get correct performance; the usage of this library must be carefully optimized to achieve correct performance.

2.5.2.2 *GPU performance*

We now compare the best CPU performance of both Caffe and Intel Caffe against the GPU implementation of Caffe (which use cuDNN and cuda). We use our K20x GPU for this evaluation.

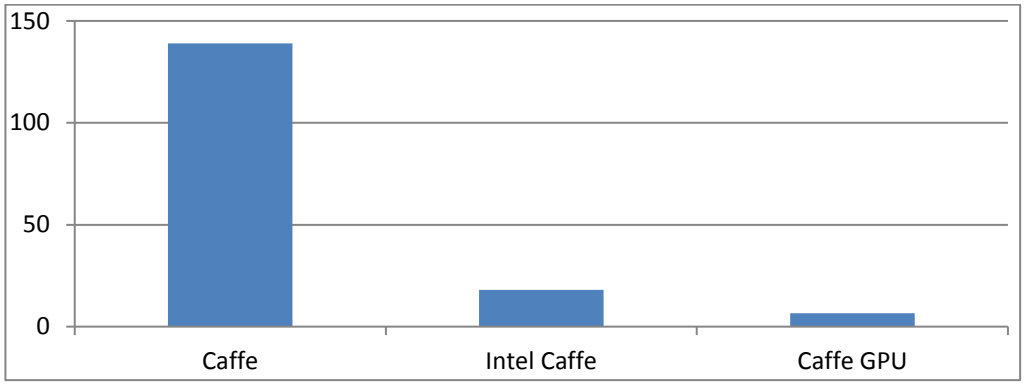

Figure 4 - Caffe CPU and GPU

Erreur ! Source du renvoi introuvable. shows that the CPU implementation cannot compete with GPU even with a highly optimized CPU implementation. Running caffe on GPU is 3X faster compared to the best CPU implementation.

2.5.3 Performance of distributed deep learning with CaffeOnSpark

We began our work on distribution by understanding the parameters that control the training. The distributed framework used in this section is CaffeOnSpark with data parallelism. In this section, we use 2 GPUs on one node.

2.5.3.1 *Distribution of the SGD with data parallelism*

Distributing the training means distributing the Stochastic Gradient Descent (SGD). To distribute the SGD we need to distribute the computations across different computing resources. To perform that task with data parallelism, we need to treat a subset of the training set on each computing resources. For example, if we have M training example and N resources, we should process $\frac{M}{N}$ training examples on each computing resource.

In practice, to process a given number of training examples, we can reduce the batch size and keeping the number of iteration constant and/or we can keep constant the batch size and reduce the number of iteration.

2.5.3.2 *Experiment 1: influence of learning rate and batch size*

The objective of the first experiment is to understand the influence of the learning rate and batch size during the training phase. Learning rate and batch size should be carefully chosen to have a good accuracy. Furthermore, when the training is distributed it's important to verify whether both the learning rate and the batch size are good. Here we show how the accuracy may change between distributed and non-distributed learning when both parameters vary.

Both **Erreur ! Source du renvoi introuvable.** and **Erreur ! Source du renvoi introuvable.** show the accuracy when the learning rate and the batch size change. For 2 GPUs we divide the batch size processed by each GPU per 2. As a result, a batch size of 25 with two GPU process the same number of training example than a batch size of 50 for 1 GPU. For this reason, we write "x2" after each batch size on figure 6.

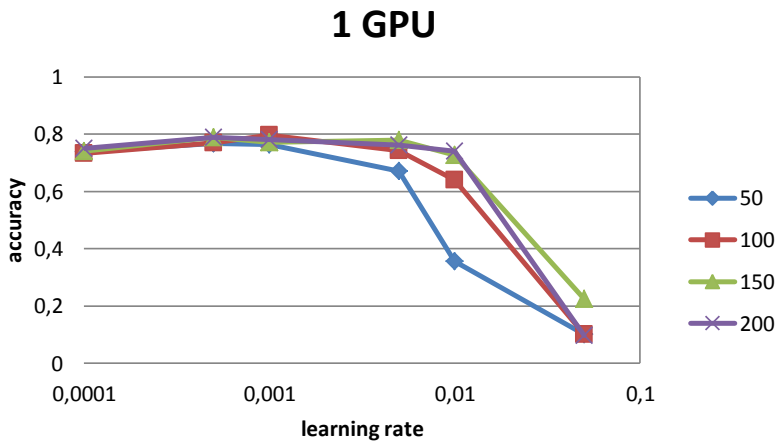

Figure 5 - 1 GPU

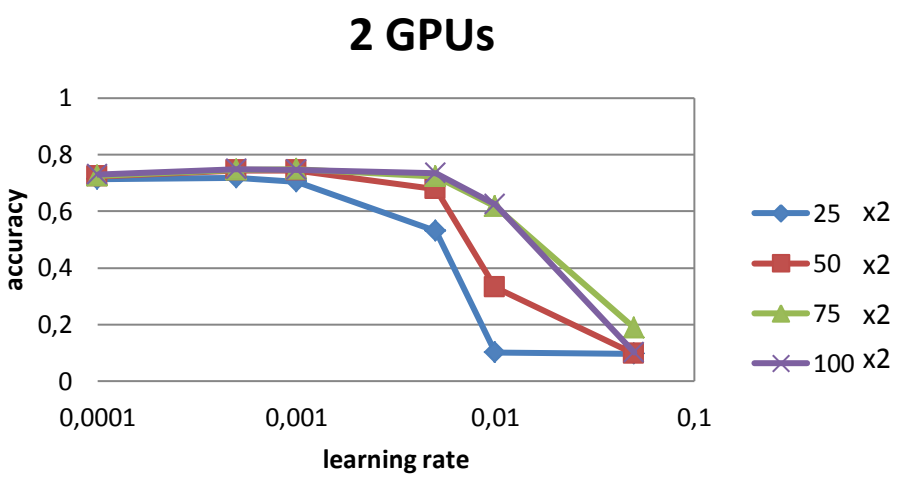

Figure 6 - 2 GPU

- Observations:

- Accuracy: we can observe that the accuracy is better with 1 GPU, as we can reach an accuracy of 80% and only 75% with two GPU

- The training is more sensitive to high learning rate with 2 GPU, especially with small batch sizes.

2.5.3.3 *Experiment 2: influence of batch size and number of iterations*

We continue our study by comparing the influence of the batch size and the number of iterations on both quality of training (accuracy) and time to train.

On these experiments we introduce the concept of epoch. One epoch consists in processing the entire training example. For example, training 1 epoch with CIFAR10 means training 50000 images, 2 epochs, 100000 images… In this experiment, we train a network from 1 to 9 epoch, we vary two parameters. Results are shown on **Erreur ! Source du renvoi introuvable.** and **Erreur ! Source du renvoi introuvable.** (the legend is the same for the two Figures).

- We vary the batch size with a fixed number of iterations (“fixed max iteration” on both **Erreur ! Source du renvoi introuvable.** and **Erreur ! Source du renvoi introuvable.**).
  For example, for 2 epochs we have a batch size of 50 and a number of iterations of 1000, and for 4 epochs we have a batch size of 100 and the number of iterations stays at 1000.
- We vary the number of iterations with a constant batch size (“fixed batch size” on both **Erreur ! Source du renvoi introuvable.** and **Erreur ! Source du renvoi introuvable.**).
  For example, for 2 epochs we have a batch size of 50 and a number of iterations of 1000 and for 4 epochs, we have a batch size of 50 and 2000 iterations.

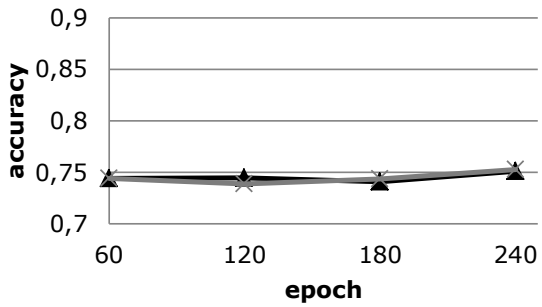

Figure 7 - Evolution of accuracy

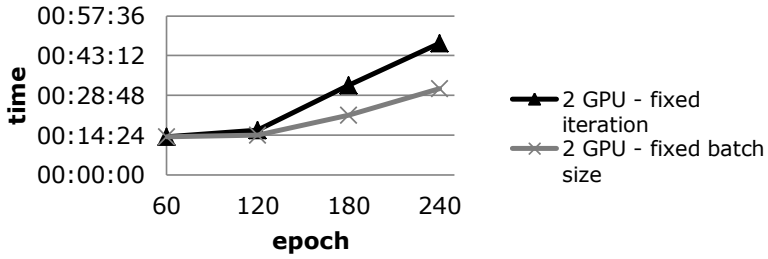

Figure 8 - Time needed to process the same number of images

- Observations:
  - Computing performance: we get better performance when the batch size is small
  - Accuracy: the accuracy is comparable for both approaches

2.5.4 Deep learning at supercomputer scale

The 2017 Nips conference took place at Long Beach in California. A workshop [9]was dedicated to deep learning at large scale and state-of-the-art results were presented. Facebook did a talk to unveil its result on a distributed training of ImageNet across more than 256 GPU with Caffe2 and a linear scalability which lasts 1 hour. Google revealed the new interconnection capabilities of its TPU deep learning accelerator and showed some figures on an ImageNet distributed training across 64 TPUs in less than 30minutes. The race is open …

[9] https://supercomputersfordl2017.github.io/

2.5.5 Conclusion

These experiments show that we need to pay attention to training parameters when distributed training is employed with data parallelism. Using the same parameters than with non-distributed learning is not possible.

The learning rate should be decreased to get the same level of accuracy when the number of iteration is fixed. Furthermore, increasing the batch size leads to computing time issues. It would be better to keep a reasonable size of batch size and increase the number of iterations to get the best performance for both accuracy and computing time

2.6 Performance Evaluation With CNTK (multi-node)

In this part, we will be working on two different nodes, each with two NVidia M60. In this section, every measure will be done on CNTK, as we were not able to make CaffeOnSpark work on two nodes.

2.6.1 Network communication

When using two different nodes, a new parameter is the network link we are using. We had two possibilities, either an Ethernet connection or InfiniBand (IB).

IB is a standard in HPC cluster because a large part of scientific simulations needs a large bandwidth or/and a small latency. The aim of this study is to understand whether IB is also useful for deep learning applications.

We made two experiments. The first test consists in increasing the number of epoch and comparing the time with a batch size set at 50 (Fig 11.a). With the second test, the batch size is set to 100 (Fig 11.b)

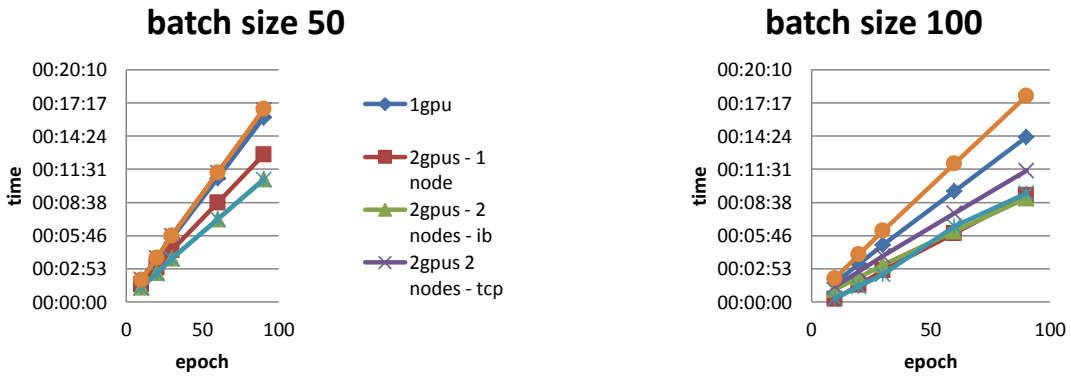

Figure 9 - a) Batch size of 50 b) Batch size of 100

These two experiments show that IB considerably improves the performance compared to Ethernet for both a batch size of 50 and 100. Compared to a batch of 50, we note with a batch size of 100,

performance between IB and Ethernet are slightly closer. This is easily explainable because when we increase the batch size, there are less communications.

2.6.2 Influence of the batch size

With the previous experiments, we may thing that increasing the batch size to a high value can reduce the communication cost and remove the advantage of IB over ethernet.
To test this hypothesis, we have chosen to measure the training error when the batch size is increased.

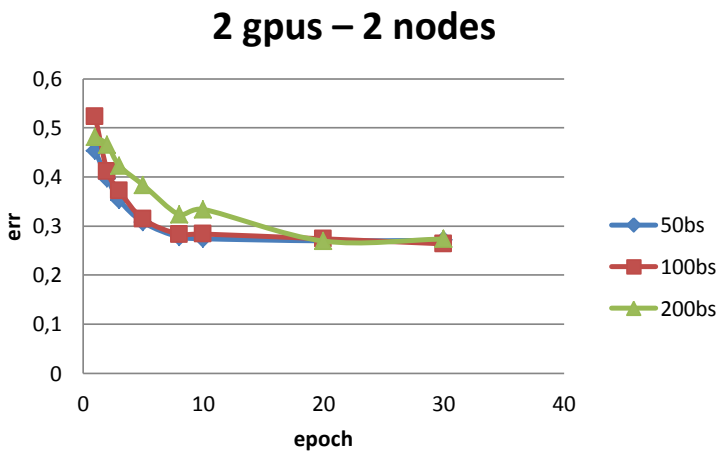

Figure 10 - Large batch size and interconnect latency

As we show, a high batch size implies a higher error. Hence, with a high batch size, more epochs is needed to reach convergence (it needs 10 epochs for a batch size of 10 and 20 epochs for a batch size of 200). Therefore, it is not possible to increase indefinitely the batch size to reduce the impact of the network: having an efficient network is essential for deep learning.

2.7 Conclusion and future works

The presented results confirm that GPU are considerably faster compared to an optimized CPU implementation on our HPC architectures. Furthermore, the SGD parameters have to be carefully chosen when training is distributed.

Deep learning is still far from having reached maturity. Many frameworks exist and each company has built its own framework to fit their particular needs.

Even on the field of distribution the situation is unclear. Data parallelism is used today but model parallelism can be a better solution if models become larger and larger because only a fraction of the parameters has to be exchanged. A low latency network will be probably required with model parallelism: we need to make experiments to confirm that. In any case, an efficient communication network will be necessary for distribution: Intel states that the Omnipath interconnect will be efficient with deep learning [6]. Atos BXI should also be competitive in this field.

Furthermore, it's even not clear that the amount of data needed today by deep learning will remain. Many researches are done to reduce the training set for deep learning[10] and other researches focus on un-supervised learning[11], which means learning without an initial labeled data set. For the moment, innovations are driven by the needs of major tech companies which own an incredible amount of data (Google, Facebook, Baidu…).

[10] http://www.geometricintelligence.com/
[11] https://users.ics.aalto.fi/juha/papers/DeepReview.pdf

3 State of the Art on data analytics and deep learning applications

3.1 Big Data Analytics

With increasing mobile broadband subscriptions [12] (3.6 billion subscriptions in 2016), mobile users may serve as a rich source of information providers. Mobile location-based services are thriving and provide an opportunity to collect fine grained spatio-temporal data of the places mobile phone users visit. This type of multi-dimensional data offers various new possibilities to tackle both new and existing research problems in the areas of human mobility. The research community is actively in pursuit of identifying and tackling these challenges. Below we give an overview of some of the work being conducted in the field.

3.1.1 Descriptive, predictive, prescriptive analysis

Blackett [13] classifies data analytics into three categories: descriptive analytics, predictive analytics and prescriptive analytics. In this taxonomy, descriptive analytics as the name suggests, "Describe" or summarize the raw data input to the system into something that could be interpreted by humans. Descriptive analytics use historical data to summarize and understand how this past event can influence any future outcomes. Descriptive analysis primarily use techniques such as data aggregations and data mining to present and describe data in a format which can be easily understood. The Predictive analytics on the other hand “Predicts” what might happen i.e., it uses various techniques to predict future trends/events. Predictive analytics provide probabilistic estimations on the likelihood of a future outcome and can provide actions that can be taken based on the data for that future outcome. The main advantage of the predictive analysis is that we can use patterns found in the historical or current data to forecast what might happen in the future including risks and opportunities. Natural language processing, data mining and text analysis approaches allow to create predictive intelligence based on the patterns and relationships among different entities in different datasets. The Prescriptive analytics focuses primarily on recommendation and assists in decision making. Prescriptive analytics can quantify the effect of future decisions and help end users to identify issues including what will happen and why it will happen. Predictive analytics can help guide the users towards finding an optimal solution based on these outcomes. In the prescriptive analysis, it is possible to analyze the feedback coming from the actions carried out using derived decision rules using decision trees, Fuzzy logic or neural networks based upon the complexity of the rules that we have to build for the requirements [14].

Authors in [15] studied and compared the lifestyle behavior of people living in cities of different sizes. They extracted salient mobility and work-rest patterns for a large population of users within each metropolitan area and show findings that quantitatively compare the behavior of people from big cities with people in smaller cities. In [16], authors investigate the mobility pattern of the user of location based social networks with special importance to the places they visit with evolution of time. They incorporate temporal regularities in their study and using their findings, cluster the users based on their behavior and predict the user’s future mobility. An overview of research activities which are conducted on the location based social networks is presented in [17]. It summarizes the work on analyzing and predicting social ties, analyzing human behavior in space and time, extracting the knowledge on user’s location.

In [18], authors perform activity forecasting in dual-agent context via image processing to predict the dynamics of human interaction. In [19], authors use the location based data provided by users of Foursquare to predict the next venue a mobile user will visit. They study the behavior of user at varying granularity and propose a set of features that capture the factors that may drive a user's mobility behavior. In [20], authors extract and model the traffic pattern of cellular towers in metropolitan cities. They use a time series analysis approach and decompose the traffic into regularity and randomness and use the prediction to forecast traffic patterns based on regularity component. Interestingly, they also show that predicting randomness component of mobile data is impossible. In [21], authors use Bayesian method to estimate the parameters of seasonal autoregressive integrated moving average (SARIMA) model. They show that forecasts from the Bayesian model are a good match to peak time traffic behavior and rapid fluctuations and use the model to predict the behavior of the traffic in Dublin. Authors in [17] revisited literature on short-term traffic forecasting and its advancements. Their findings support the research interest towards:

i. Responsive forecasting schemes for non-recurrent conditions
ii. Developing prediction systems with increased algorithmic complexity
iii. Attempting to understand data coming from novel technologies and fuse multi-source traffic data to improve predictions
iv. The applicability of AI methodologies to the short-term traffic prediction problem.

In [22], authors state that current data analytics do not fully exploit the data generated during execution of processes while continuous improvements of such process is a key to achieve success in companies. They provide an insight that the existing techniques do not consider prescriptive analysis that could transform the results into improved actions. The authors propose a recommendation based business process optimization driven by data mining approach to prescriptively generate action recommendations. In [23], authors provide a prescriptive analytic system that provides advice to researchers about their future research direction and strategies. The system called InSciTe uses various data sources including but not limited to papers, reports, patents, etc. and provides advices and also a group of role model researchers and how to be like them. Authors in [18], propose an adaptive middleware concept that enables distributed computation on the enterprise level and on the field level to support power systems. They use linked data in their model to provide a map for moving the computation to the data required for analysis. The idea is to move the computation closer to the data source to minimize congestion and improve overall performance of the big data system. Whereas authors in [24] provide a literature survey by analyzing the research related to big data analytics. They carefully categorize the research in big data and observe that volume, velocity and variety of data is increasing and the heterogeneous and distributed nature of these data pose a lot of challenges to the big data analytics.

3.1.2 Big data analytics techniques and tools

In these different analysis methods, datasets can be obtained from; unstructured data sources such as social media content or data streams that provide large number of textual attributes, structured data sources that depend on predefined data models created for storing, processing and accessing information such as age, gender, revenue etc. and semi structured data sources such as XML or JSON. In the majority of cases, unprocessed datasets, particularly big data, do not provide powerful insights.

Yet, by applying the right techniques and tools to perform data reduction, we start to see different patterns that lead towards future predictions [25].

3.1.2.1 *Big data analytics tools*

At present, it may not be possible to process the whole dataset obtained from heterogeneous sources at once; thus, adapting a good analytical framework is important for the data analysis process.

Big data processing frameworks can be found as both commercial and non-commercial and they are capable of processing data in realtime or by performing batch processing. The popular open-source bigdata processing frameworks includes Hadoop, Spark, Flink, Storm, and Samza. Hadoop[12] is quite a simple framework compared to the other frameworks. It can process data in batch, and splits into smaller processing jobs, spreads across a cluster, and recombine their results. Hadoop can use MapReduce as a service to write programs to perform computations on data in parallel and in a distributed fashion. It is supported for several scripting languages including Pig, Hive, Flume and HDFS.

The Hadoop framework can accommodate the Spark engine[13] in place of MapReduce. Spark does not include its own distributed storage layer, and as such it may take advantage of Hadoop's distributed filesystem (HDFS), among other technologies unrelated to Hadoop. Spark differs from Hadoop and the MapReduce paradigm in that it works in-memory, speeding up processing times.

Flink[14] is a streaming dataflow engine, aiming to provide facilities for distributed computation over streams of data. Flink can perform real-time data processing and also batch processing with the use of several APIs, including a streaming API for Java and Scala, a static data API for Java, Scala, and Python, and an SQL-like query API for embedding in Java and Scala code. The most important concern is that it has its own machine learning and graph processing libraries.

Apache Storm[15] can be used for real-time analytics, distributed machine learning, and numerous other cases, especially those of high data velocity. Storm can run on YARN and integrate into Hadoop ecosystems, providing existing implementations a solution for real-time stream processing. Storm is designed for easily processing unbounded streams, and can be used with any programming language.

Turi[16] is another open-source based toolkit originally developed for Machine Learning tasks, it has found a great success at a broad range in other data-mining tasks.

3.1.2.2 *Big data analysis techniques*

Data analysis enables an organization to handle abundant information that can affect the business. Data analysis has two main objectives: to understand the relationships among features and to develop effective methods of data mining that can accurately predict future observations [26]. Available analytical techniques include data mining, statistical analysis, and machine learning.

[12] http://hadoop.apache.org/
[13] https://spark.apache.org/
[14] https://flink.apache.org/
[15] http://storm.apache.org/
[16] https://turi.com/

Data mining is a process for extracting hidden, unknown, but potentially useful information and knowledge from massive, incomplete, noisy, fuzzy, and random data. Most influential data mining algorithms include C4.5, k-means, SVM, Apriori, EM, Naive Bayes that perform classification, clustering, regression, statistical learning, association analysis, and linking mining.

Cluster analysis or clustering is the task of grouping a set of objects in such a way that objects in the same group are more similar to each other than to those in other groups. It differentiates objects with particular features and distributes them into sets accordingly. Cluster analysis is an unsupervised research method that does not use training annotations.

Statistical analysis is based on the statistical theory, a branch of applied mathematics. In statistical theory, randomness and uncertainty are modeled with Probability Theory. Statistical analysis can provide a description and an inference for big data. Descriptive statistical analysis can summarize and describe datasets, while inferential statistical analysis can draw conclusions from data subject to random variations. Statistical analysis is widely applied in the economic and medical care fields. Traditional methods for statistical analysis e.g., sampling, interpreting results, have been used by scientists for thousands of years. But high data volume make statistics ever more valuable with powerful computers and advanced algorithms that have all led to an increased use of computational statistics.

Correlation analysis is a method of statistical evaluation used to study the strength of a relationship between two, numerically measured, continuous variables. If correlation is found between two variables it means that when there is a systematic change in one variable, there is also a systematic change in the other. The output of correlation analysis can be positive or negative. Positive correlation exists if one variable increases simultaneously with the other and negative correlation exists if one variable decreases when the other increases.

Regression analysis is a mathematical technique that can reveal correlations between one variable and others. It identifies dependent relationships among randomly hidden variables on the basis of experiments or observations. Regression analysis helps to understand how the typical value of the dependent variable changes when any one of the independent variables is varied, while the other independent variables are held fixed.

In data analysis three types of machine learning algorithms are mainly used: supervised learning, unsupervised learning and reinforcement learning. Supervised algorithms consist of a target / outcome variable (or dependent variable) which is to be predicted from a given set of predictors (independent variables). In unsupervised learning algorithms, we do not have any target or outcome variable to predict / estimate. This often boils down clustering population in different groups, which is widely used for segmenting customers in different groups for specific intervention. In reinforcement learning algorithms, the machine is trained to make decisions that maximize a long-term reward. This machine learns from past experience and tries to capture the best possible knowledge to make accurate business decisions. Many types of machine leaning algorithms exist within each of these three broad classes of methods: linear regression, logistic regression, decision trees, support vector machines, Naive Bayes, k-nearest neighbors, random forest, k-means…

3.1.2.3 *Text analytics and pattern recognition*

Text analytics (text mining) refers to techniques that extract information from textual data. Social network feeds, emails, blogs, online forums, survey responses, corporate documents, news, and call center logs are examples of textual data held by organizations. Text analytics involve statistical analysis, computational linguistics, and machine learning. Text analytics enable businesses to convert large volumes of human generated text into meaningful summaries, which support evidence-based decision-making. There exists multiple text analytics methods such text summarization, information extraction, sentiment analysis, and question and answering.

The technique of extracting structured data from unstructured or semi-structured sources is called information extraction. The specific type and structure of information extraction varies depending on the application, such as in biomedical research, decision making for finance companies etc. The most important tasks for information extraction are named entity recognition (anything that can be referred to with a proper name: person, location, organization…) and relation extraction [40]. Typically, in information extraction a set of sub tasks have to be done including:

i) Pre-processing of the text (this is where the text is prepared for processing with the help of computational linguistic tools such as tokenization, sentence splitting, morphological analysis, etc.)
ii) Classifications (mentions of people, things, locations, events and other pre-specified types of concepts are detected and classified.)
iii) Connecting the concepts (task of identifying relationships between the extracted concepts.)
iv) Unifying (presenting the extracted data into a standard form.)
v) Noise removal (eliminating duplicate data.) [27].

Text summarization techniques automatically produce a succinct summary of a single or multiple documents. The resulting summary conveys the key information in the original texts. Applications include scientific and news articles, advertisements, emails, and blogs. Broadly speaking, summarization follows two approaches: the extractive approach and the abstractive approach. In extractive summarization, a summary is created from the original text units (usually sentences). The resulting summary is a subset of the original document. Based on the extractive approach, formulating a summary involves determining the salient units of a text and stringing them together. The importance of the text units may be evaluated by typically analyzing their location and frequency in the text. Extractive summarization techniques may not really require an ‘understanding’ of the text. In contrast, abstractive summarization techniques involve extracting semantic information from the text. The summaries contain text units that are not necessarily present in the original text. In order to parse the original text and generate the summary, abstractive summarization incorporates advanced Natural Language Processing (NLP) techniques [28].

3.2 Natural Language Processing

Natural Language Processing (NLP) is a broad term that encompasses many research domains that propose methods and algorithms to automatically process natural language, either as input to

compute useful information from text and speech, or as output to generate meaningful and human-like sentences and documents. NLP includes research domains such as automatic speech recognition, automatic translation, conversational agents, document summarization, and so on. We focus next on three areas that are particularly relevant for the PAPUD project: sentiment analysis, relation extraction and topic modeling, but we first present in Sections 3.2.1 and 3.2.2 a general overview of classical vs. deep learning methods applied to Natural Language Processing.

3.2.1 Non-deep learning NLP

NLP and text analysis approaches can be divided into linguistic, semantic, statistical and hybrid, which combine two or more of the previously mentioned [29]. Although less common as they require linguistic knowledge, linguistic methods can provide a deeper understanding of the content of a given text, but fail at processing vast amounts of data without the use of procedural methods. Linguistic techniques take into account the language characteristics of the text, such as syntax and grammar [30]. Syntax analysis belongs to these, usually employing *Part-of-speech tagging* techniques, which tag words by their grammatical function based on definition and context [29].

Semantic methods are closely related to linguistics, as the meaning of a text depends on its structure. To analyse semantic data, several approaches and algorithms are discussed in [31]. A common method is *Latent Semantic Indexing*, which reduces data dimension and tackles the problems related to polysemy and synonymy. Some approaches use verbs in order to identify relations on the text, but solutions are usually domain and/or language dependent. For a Natural Language Processing approach we can find algorithms such as *Named Entity Recognition,* which classifies named entities into program-defined categories, and *Semantic Role Labelling,* which assigns semantic roles to words or phrases in a sentence.

Finally, statistical approaches make use of already established machine learning algorithms to extract information from text. Some of the most popular statistical methods are *Logistic Regression*, *Support Vector Machines* and *Decision Trees* discussed in Section 3.1.2.2. These algorithms are trained by a classified subset of text and tested on an unclassified one [32].

3.2.2 Deep Learning methods applied to NLP

In recent years deep learning methods gained considerable attention in text mining and Natural Language Processing (NLP) domain ([33], [34], [35], [36], [37], [38], [39], [40], [41], [42], [43]) and bioinformatics ([44], [45], [46]). The fundamental idea behind deep learning algorithm is automating the mining of representations from data using a hierarchical layered scheme instead of a single layer scheme ([47]).

Unsupervised machine learning models usually exploit generative models. In the deep learning domain, two such generative models include auto-encoders and Restricted Boltzmann Machines (RBMs). Auto-encoders ([48]) are deep learning networks constructed of 3 layers: input, hidden and output. Auto-encoders attempt to learn the representations of the input in the hidden layer, and then try to reconstruct the input in the output layer based on these intermediate representations. A basic auto-encoder learns its parameters by minimizing the reconstruction error. This minimization is usually done by stochastic gradient descent with regularization like “regularized auto-encoders” as in ([49]). Deep Boltzmann Machine (DBM) are generative models with multiple RBMs stacked together ([50]); it

is a special case of the general Boltzmann machine (BM). In a DBM, each layer captures complicated, higher-order correlations between the activities of hidden features in the layer below.

Auto-encoders may be used for text classification and text generation. For instance, Semeniuta et al. ([51]) suggest a new composite model that combines fully feed-forward convolutional and deconvolutional components with a recurrent language model for text generation. One of the most attractive properties of their architecture is being more effective at forcing the decoder to make use of latent vectors. According to their experiments on the Penn Treebank dataset ([52]) and one million random tweets collected using the Twitter API; they observe the effectiveness of their model especially on long sequences. In another study, the authors extend sequence-to-sequence Long Short Term Memory (LSTM) auto-encoder models, which do not consider the intrinsic hierarchical discourse structure of texts, for multisentence generation ([53]). They state that their proposed hierarchical LSTM models can partially preserve the semantic and syntactic integrity of multi-text units and generate meaningful and grammatical sentences in coherent order. Their model performs better than standard sequence-to-sequence models based on their experimental results ([53]).

Hinton and Salakhutdinov ([34]) present a generative deep learning model to acquire knowledge of the binary codes in documents. There are many layers in their deep learning network, where the lowest layer shows standard term frequency vector of the document and the highest layer shows the acquired binary code of the document. It is demonstrated that the binary codes of the documents that are semantically similar place comparatively nearer in the Hamming space ([34]). According to the experimental results, using these binary codes in a deep learning architecture for document retrieval generates higher accuracy in a faster way compared to the standard term-frequency vector representation in semantic-based analysis.

Ranzato and Szummer ([54]) present a semi-supervised deep learning model. In this study, they offer an algorithm to learn text document representations based on semi-supervised auto-encoders that are stacked to form a deep network. The model can be trained efficiently on partially labeled corpora, producing very compact representations of documents, while retaining as much class information and joint word statistics as possible. The authors show that for learning compact representations, deep learning models are better than shallow learning models because they need less computations and storage capacity.

Interacting with computers in natural language requires a representation of words, their meaning, and their meaning in context with other words. A recent development towards that goal is Vector Space Models (VSM) for words. VSMs represent each word by a high-dimensional vector, which is learned automatically from a large unannotated natural language corpus. Mikolov et al. ([36] [37]) create a neural-network based model for automatically learning word vectors based on the distributional hypothesis ([55]) which says that ”you shall know a word by the company it keeps”. This model and its open source software word2vec is a very popular example and has received important attention in the scientific community within a short period with several extensions and improvements being published. Word2vec is actually an unsupervised NLP tool, which uses artificial neural network structure. The tool receives a text as input and represents each word in the text as a vector. Word2vec clusters semantically similar words in close coordinates. In order to find the coordinates of the words two

learning algorithms are used, namely Continuous Bag of Words (CBOW) and skip-gram (SG). In the CBOW architecture, a word is looked at with left and right neighboring words in a fixed-length window, and is attempted to be estimated from its neighboring words. SG architecture works exactly in the opposite way; it tries to predict the right and left surrounding words of a specific word by just looking at the word itself. In general, the SG architecture produces better word vectors for infrequent words while CBOW produces better results on a larger corpus. CBOW and SG may use one of two learning algorithms, namely hierarchical softmax (HS) and negative sampling (NS). Generally, HS training algorithm produces good results for infrequent words while NS produces better results for frequent words. One of the greatest properties of word2vec is that it maps some arithmetic operations between word vectors in the embeddings space to interpretable semantic transformation operations.

Since word2vec learns from the context of a word, i.e., where it is usually `embedded' in a text, it is also known as Word Embeddings. Embeddings have been used to improve NLP tasks such as NER, Part Of Speech (POS) Tagging, Dependency Parsing, and text classification.

Word embedding algorithms are able to extract semantic relations from very large amounts of textual documents generally with the help of feed-forward artificial neural networks. However, they need a great amount of textual data to perform reasonably well. Studies have been carried out to improve the semantic relations between the words that these models reveal by using external semantic sources. In one of these studies, Wikipedia is used as an external semantic source, aiming to better integrate word2vec-like distributed approaches to problems involving small amount of labeled data ([56]). In another study ([41]), an n-gram and word-based artificial neural network is presented to model topics similar to the general topic modeling method with Latent Dirichlet Allocation (LDA) ([57]). This unsupervised method has achieved high accuracy among the text mining algorithms used for topic identification/extraction. Another study, labels of semantic features, which were produced by human experts, were used to solve the problem of requiring very large training set requirement of word embedding algorithms such as word2vec and to work well in small data sets ([38]).

In a recent study, Word2vec was used as a preliminary method with the hypothesis that it could improve the performance of classification by extracting semantic relations before text classification ([40]). This is compared with the tf-idf word weighting method, which is a widely used unsupervised weighting method. In this study, the skip-gram approach in the Word2vec package was trained with a default vector length of 100, and the sum of weighted word vectors according to their frequency in the document is used for the representation of documents.

3.2.3 Sentiment analysis & opinion mining

Sentiment analysis, opinion mining and argumentation mining are all related domains that analyze some input text or speech to respectively compute the sentiment of a paragraph, sentence or phrase, the opinion of the author about some specific aspect or topic, and the arguments used by the author to support his opinion.

Sentiment analysis has been the subject of a huge amount of works and publications in the last ten years, due to its easy application to several consumer-based services, such as analyzing the impact of advertisement on consumers, evaluating hotel reviews or even financial forecasting [58]. Since a few years, nearly every competitive sentiment analysis model exploits deep learning models, either based

on shallow convolutional networks, such as the famous award-winning Yoon Kim model [59], or exploiting specific recurrent architectures to accomplish a given task, for example target-based sentiment analysis, which model has been presented at the SemEval 2017 competition [60]. The SemEval evaluation campaign [61] is a reference international campaign for the evaluation of sentiment analysis systems since 2007. It proposes various tasks every year to assess the quality of sentiment analysis technologies in various conditions: from simple message polarity classification (inferring whether a message is globally positive, negative or neutral) to measuring the sentiment on a five-point scale about a predefined topic. Several other shared tasks have also been organized in this domain [62]. Beyond positive and negative sentiments, more complex emotions are also often considered (e.g., fear, anger, worry, relief) with varying emotion granularity and strength.

Whereas most sentiment analysis research focuses on identifying explicit sentiment expressed by linguistic patterns, there is only limited research aiming at detecting implicit sentiment (e.g., [63] [64]). In these cases, the sentiment needs to be inferred through commonsense knowledge. While, recent work has focused on the identification of implicit sentiment at the coarse-grain event level [65], it would be interesting to capture contextual commonsense knowledge with deep learning approaches and as such refine the detection of implicit sentiments.

Finally, to find out the underlying reasons why opinions are expressed, we will explore a machine learning method based on capturing deep features to detect the arguments (premises and conclusions) in a document, the relations linking them and the internal structure of each argument. Computational argumentation is a new field of research ([66], [67]) and there are various ways to determine the criteria which contribute to good arguments from text. End-to-end deep learning approaches [68] which prevent error propagation or joint modelling approaches that incorporate several tasks during training could possibly lead to an argumentation structure parser with the ability to even determine flawed argumentation structures. Another field of research which can be used for argumentation mining is structure relational learning where recently [69] have shown the benefits of using factor graphs to extract argumentation structures. Research in capturing relational and structured dependencies between recognized classes in the deep learning paradigm only starts to emerge. We will investigate structured recognition for argument mining in deep learning frameworks including long short-term memory networks ([70]) that learn to align the components of argumentation possibly in a multi-task setting ([71]). Although argumentation modelling is very challenging, we are strongly convinced of its value as it allows to not only detect the target (*what*) of the sentiment, but also its reason (*why*). Again, it would be interesting to capture argumentative context (i.e., macro-level information) and captures this by deep learning approaches in order to identify argumentative relations in a more accurate way. [70] realize this by means of stacked classifiers, but not in a deep learning fashion.

All the above approaches rely on supervised learning that is learning from annotated training examples. Especially, deep learning approaches demand sufficient amounts of training examples in order to learn accurate models. When building systems in different languages this is often a problem. It is often the case that sufficient annotated training data are available in one language, but very few or no training data are available in other languages. Hence, we will investigate in PAPUD how to transfer and adapt models trained for one language to another language for which few annotated

training data are available. [72] learns interlingual sentence representations from parallel data. [73] jointly learns sentence representations across six very different languages in a rather conventional neural machine translation set-up. Successful approaches of domain transfer learning today rely on effective multi-task learning for sentence representations that combines the inductive biases of diverse training objectives in a single model [74]. Such models can be trained on several data sources. We will extend these models to transfer between different languages using several sources (e.g., parallel data for learning a translation model, as well as supervised objectives in one language) and model the transfer with suitable training objectives. We will investigate how to build models that jointly learn translations and supervised recognition tasks such as the ones described above when few training data in the target language are available.

3.2.4 Relation extraction

Relation extraction is an important research track at the intersection between NLP (machine reading), information retrieval (text mining) and the semantic web, which aims at inferring typed relations between named entities that occur in text documents [75]. A typical such relation is for instance:

founder-of (Bill Gates, Microsoft)

Relation extraction methods can be classified into three categories [26]: (1) Knowledge-based methods ([27], [28]), which are used usually in domain-specific tasks relying on hand-crafted pattern matching rules. (2) Supervised methods ([29], [30]), which automatically learn extractors for relations by using machine-learning techniques based on a set of tagged examples. (3) Distant supervision methods ([31], [32]), which automatically generate large amounts of training data.

Recently, more and more researchers have applied deep learning techniques in relation extraction and achieved good performance. These methods using only word embeddings as input, can extract hidden structures from relation mentions automatically, and capture global information over long distance, while reducing the workload of hand-crafted feature engineering. In the earlier deep learning works, relation extraction is modeled as a multi-class classification problem which assigns a relation class to each mention pair in the text. [76] proposes a simple end-to-end CNN network consisting of an embedding layer, a convolutional kernel layer, a dense layer and a softmax layer to automatically learn features instead of using hand-craft features. The model uses synonym vectors, and incorporates some lexical features using word lists, POS lists and entity type lists. It outperforms the state-of-the-art kernel-based model at the time on the ACE 2005 dataset by 9 points of F-score. In a more recent work, [77] proposes a max-pooling layer over the output of the convolutional network. It is the first work which uses positional embeddings together with some traditional lexical features and gave almost 9% improvement in their F-score. After that, [78] completely get rid of external lexical features to enrich the representation of the input sentence and let the CNN learn the required features itself. Their architecture consisted of word and positional embeddings followed by convolution and max-pooling. Additionally, they also incorporate convolutional kernels of varying window sizes to capture wider ranges of n-gram features.

As for sentiment analysis, current state-of-the-art relation extraction methods mainly rely on the same two types of deep neural networks: convolutional and recurrent models [79]. The following model [80] is a typical convolutional architecture used for relation extraction:

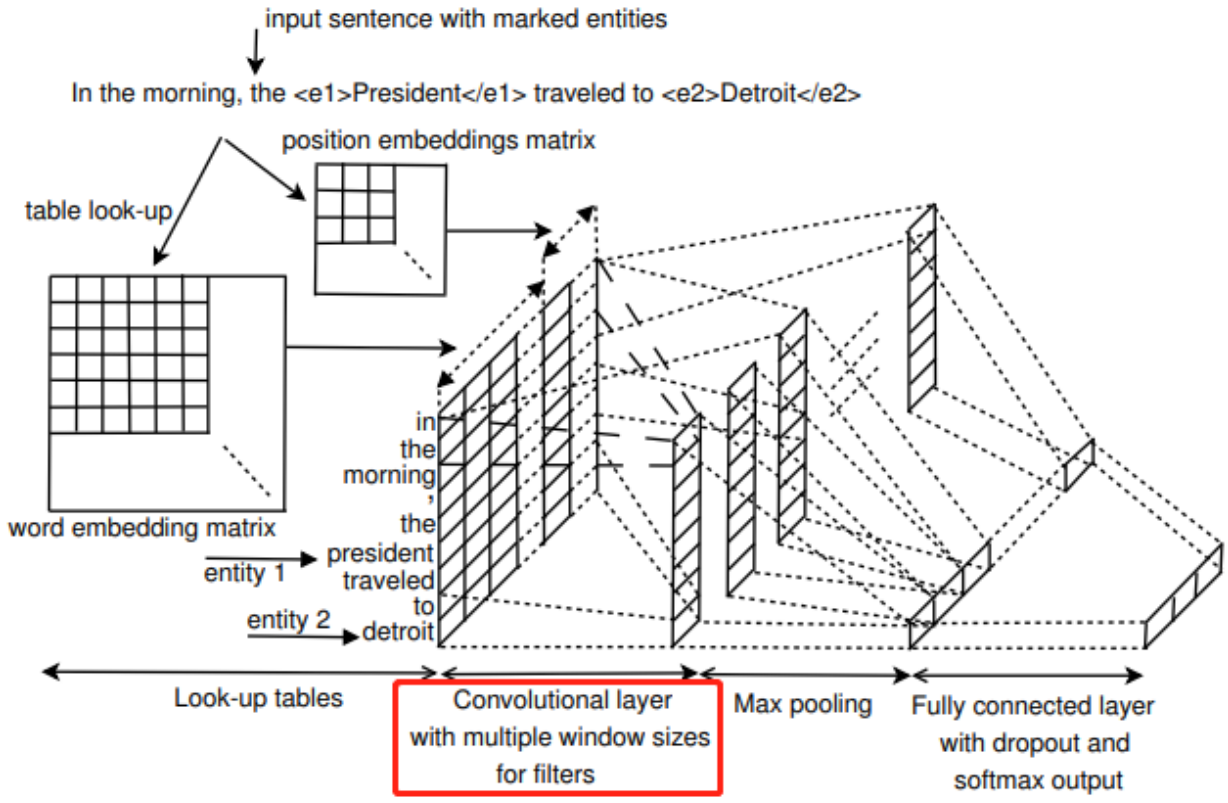

Figure 111 - A typical convolutional architecture used for relation extraction

However, interesting models occur when relation extraction between pairs of entities is encompassed within more complex hierarchical structures, such as with recursive neural networks [81].

To deal with the lack of explicitly annotated text, [82] propose a distant supervision approach for automatically generating large amounts of training based on the assumption that "if two entities participate in a relation, all sentences that mention these entities express that relation". The authors tackle the task of extracting relations of Freebase by using existing relations in Freebase as training data. For each of related pair of entities, they collect all sentences that mention both entities as input observation and use their relation type in Freebase as label. Together with a set of unrelated pairs of entities as negative examples, they train a classifier to predict relations. It is easy to see that the used distant supervision assumption can be violated. Thus, [83] relax the distant supervision assumption by modeling the problem as a multi-instance learning problem, which is a form of supervised learning where a label is given to a bag of instances, rather than a single instance. In the context of relation extraction, every entity pair defines a bag and the bag consists of all the sentences that contain the mention of the entity pair. Instead of giving a relation class label to every sentence, a label is instead given to each bag of the relation entity. There have been several distant supervision approaches using deep learning techniques. [78] proposes Piecewise Convolutional Neural Networks (PCNNs) which

used the multi-instance learning paradigm, with a neural network model to build a relation extractor using distant supervision data. The model has a similar architecture as [77], [78], with one important contribution of using piecewise max-pooling across the sentence. The authors claim that the max-pooling layer drastically reduces the size of the hidden layer and is also not sufficient to capture the structure between the entities in the sentence. This can be avoided by max-pooling in different segments of the sentence instead of the entire sentence. [84] uses an attention mechanism over all the instances in the bag for the multi-instance problem. In this model, each sentence of a bag was first encoded into a vector representation, using a PCNN or a CNN. Then the final vector representation for the bag of sentences can be found by taking an attention-weighted average of all the sentence vectors in the bag.

The current growing importance of deep reinforcement learning methods has also a profound impact on the relation extraction field [85] to enable training with weak supervision.

3.2.5 Topic modeling

Topic classification is a special case of text classification, which in general aims at automatically annotating a document with a given class label [86]. In text classification, there is no restriction on what this label represents: it may be a sentiment, the language with which the text is written, the author or the date of the document, etc. But most commonly, this label represents the topic of the text (for instance, one topic amongst *politics, sports, science, economy*, etc.), which corresponds to the topic classification task. A dominant class of approaches to achieve this task has long been Bayesian models, which are thus often called *topic models* in this context. We review in Sections 3.2.5.1 and 3.2.5.2 these specific types of models and how they are nowadays combined with deep learning.

3.2.5.1 *Monolingual models*

Recently, there has been an interest to combine traditional topic models with deep learning. We can identify two aspects in which deep learning has shown potential to enhance topic models. A first aspect is the use of neural networks to estimate the posterior distribution of the latent variables in topic models. For many topic models (e.g., LDA) exact computation of the posterior is intractable and approximate inference algorithms such as Gibbs sampling and mean-field variational inference are still computationally expensive and require the derivation of (model-specific) inference formula's, which can be cumbersome and time-consuming. Recent works [87] [88] [89] have proposed variational auto-encoders [90] as an alternative inference method, where the posterior over the latent variables is estimated by a neural network. The advantages of this approach are that a) it is fast ([88] report a training time of 80 minutes for fitting a topic model on 1 million documents on a modern GPU); and b) that the inference network makes little assumptions about the topic model. The latter enabled [88]**Erreur ! Source du renvoi introuvable.** to propose a variation of LDA, which obtains better topic coherence scores, without deriving new inference formula's. A second aspect is leveraging the predictive power that neural networks bring in the design of the generative model. Several works have proposed methods that incorporate word embeddings into topic models, e.g., lda2vec combines the objective of continuous Skip-gram model with LDA [91]. Other works have looked at getting rid of the bag-of-words assumption inherent to many topic models: in latent LSTM allocation [92], topics are generated from categorical distributions that are computed from the hidden states of an LSTM; TopicRNN is an autoencoder, where the decoder is a language model that combines a recurrent neural

network (RNN) with a topic model. The RNN aims to capture the local context of word sequences, while the topic model has to ensure semantic consistency on document-level [93]. The encoder can be used to obtain document representations which obtain performance competitive with the state of the art when used as features for sentiment analysis.

3.2.5.2 *Cross-lingual models*

Topic models have been used to induce cross-lingual space with a limited amount of supervision: both documents and words can be represented with topic distributions. When the topics are shared across languages these representations are cross-lingual. In the literature, we can distinguish between two main approaches to learn these cross-lingual spaces that differ in the bilingual signal they use: [94], [95], [96] leverage a collection of topic-aligned document pairs. A good example of topic-aligned documents are Wikipedia documents that discuss the same topic(s) but are written in different languages. The second line of research uses a bilingual seed dictionary to construct the cross-lingual space [97]. The recent advances in monolingual topic modeling naturally enable the development of new cross-lingual variants. For instance for the variational autoencoder, the encoder could be constrained to incorporate information of the bilingual seed dictionary or document-aligned comparable corpora by adding regularization terms in the training objective.

3.3 Text Analytics

Text analytics is a sub-field of big data analytics, which has been reviewed previously. Text analytics also has a large overlap with NLP methods, but while NLP aims at processing texts from the language point of view, text analytics rather adopts the information retrieval point of view. We will focus next on the sub-domains of Text analytics that have not be covered so far in this document. Text analytics is utilized in several applications of different business sectors such as financial markets, information technology, industry, health care, government and so on. In the scope of the Papud project, this section presents popularity detection and user log file analysis, which will take advantage of text analytics using deep learning techniques.

3.3.1 Text-based popularity detection

Once a content is published on a social network, it attracts different amount of user's interactions depending on its interestingness, topic, publisher's reputation, publishing time and etc. [98] [99]. Meanwhile, some contents are succeeded to attract more user engagements and become popular [100]. Popularity of a content is usually assessed by different cascading metrics such as number of likes, shares, views, etc. Predicting the trend of popularity for a content (which can be a text, video, or image) and more importantly identifying the users who are going to react to that content are very valuable information for different entities such as service providers to rank the content better [101], to discover the trending posts early, to improve recommendations and even to improve their content delivery networks and user experiences [102]. Authors in [103] have proposed a tree-structured long short-term memory network to learn and predict the entire diffusion path of an image in a social network. In [104], authors investigate the sequential prediction of popularity by proposing a prediction framework, by incorporating temporal context and temporal attention into account.

Text-based popularity prediction for tweets is done using dynamic multimodal regression in [105]. They demonstrate that content based features can be used to improve upon social features and

dynamics features via their joint-embedding regression model. Authors in [106] investigate the impact of textual features on popularity of user posts on Instagram. They show that visual and textual features are complementary in predicting the popularity of a post. An approach for predicting the popularity of user posts by considering preferences of individual users to the items is presented in [107]. They factorize the popularity of posts to the user-item-context and propose a multimodal context-aware recommender. The proposed approach has the ability to predict the popularity of posts related to different items which are shared by a specific user. Moreover, it is able to predict the popularity of posts shared with different users for a specific item.

3.3.2 User log file analysis

System log files obtained when running various systems mostly contain information in a text format, so that they can be read and understood by human experts. This is a very specific type of text, but a large part of it includes natural language fragments of sentences, usually in technical English. Automatically extracting information from these log files is a sub-area of text analytics called log analysis. Log analysis pertaining to a failed run can be a tedious task. Particularly, if logs contain several parameters. Indeed, this task deals with finding text which can provide some useful information to understand the reason of failure of a run. This information is always required for online monitoring and anomaly detection.

Due to its importance, several approaches have been proposed to predict anomalies using information extracted from logs. According to [108], these approaches can be classified in three main categories: PCA based approaches over log message counters [109], invariant mining-based methods to capture co-occurrence patterns between different log keys [110], and workflow-based methods to identify execution anomalies in program logic flows [111]. These approaches already showed some limitations. The PAPUD project will use user log file analysis in BAREM use case which will be discussed more in section 4.5.1.

3.3.2.1 *Deep Learning for log analysis:*

Recently, Deep Learning (DL) demonstrated very promising result for log analysis. Several sophisticated architectures were built on top of DL based approaches. The authors of [112] propose a sophisticated system named DeepLog. First, the system identifies concepts that follow certain patterns using a combination of natural langue processing tools. Then, a multilayer neural network with Long Short-Term Memory (LSTM) is used as prediction system. DeepLog is incremental, which means that the system can adapt to new log patterns that emerge over time.

Authors of [113] propose a neural network language model learned from ‘successful’ log files, to predict the anomalies in a ‘failed’ run, thereby attempting to reduce the size of the log file which needs manual review. The architecture of the proposed system is built on top of word vector based RNN language model as well as char vector-based language RNN model as described in [113]. The experimental results show the benefit of using deep neuronal network for anomaly detection.

Several other existing approaches based on Deep Learning have been proposed for log analysis [114] [115]. All of them show very interesting results compared to previous approaches.

3.4 Online Social Networks Data Analytics

Users on OSNs generate a huge amount of data including contents and actions. Performing analytics of this data will help different user applications to model user behaviors and improve the provided services and user experiences. This section explores the deep learning techniques to analyze OSN users' activities, detect users' communities, and analyze the influence and link between users. Also, deep learning methods to process user-generated text and media and collaborative recommendations are discussed.

3.4.1 User activities analysis

User activities on social media constitute potentially rich resource for various objectives and applications such as recommender systems, popularity prediction, and understanding user behaviors. Collaborative filtering is one of the methods that utilizes user activities information in order to provide recommendation to users. Item2Vec [116] is a deep learning method which uses user interest list to extract items embedding. In [117], authors predict popularity of an image through online social media exploiting the image content and some social cues of user activities such as number of photos uploaded by a user or number of contacts a user has. The results show interesting variation in importance of those social cues.

Authors in [118] track user's activity patterns in long and short sessions. These activities include user's likes and comments. They found that longer sessions have different characteristics than shorter ones, and this distinction is already visible in the first minute of a person’s session activity. This allows the prediction of the ultimate length of his or her session and how much content the person will see. The future popularity of cold-start news is predicted using user activity in [119]. The authors used number of likes, shares, and comments of users as metrics and studied their correlation to the final popularity of cold-start news. In [120] user activities are used to extract user embedding by introducing user2vec model. This model extracts user embedding from her activity log and use those embeddings to predict user's future activity.

3.4.2 Community Detection

Communities are an integral part of online social networks and represent user groups based on the connections between them. Community detection is a data clustering problem which aims to assign each node or user to a community, and it allows to predict connections not yet observed or infer information about social network users [121]. These structures are expressed by a subgraph of the graph that represents the whole social network. Due to the nature of these services several communities can overlap with each other, which needs to be taken into account by the analysis algorithms [122]. The nature of these graphs, often unweighted, with high edge density and an unknown number of communities makes it quite difficult to detect the finer structure of the network, usually leaving the smaller communities out of the output of algorithms [123].

One of the best approaches to this problem the *Leader-follower algorithm*, which classifies the users in "leaders" (nodes which connect to different communities) and loyal followers (nodes which only have neighbours within a single community) to identify communities [123]. This approach doesn't require prior knowledge of the number of communities in the network and is able to detect overlapping communities [124]. The nature of social network structures also makes it one of the most

accurate community detection algorithm for these services. A faster alternative named *Fast Leader-Follower Algorithm* has recently been proposed [124].

3.4.3 Influence Analysis

Influence is the effect that certain social network users have over others' opinion or available information on a topic. It is then important to identify the most influential or trusted users (influencers) as their opinions are the widest and fastest reaching on social networks [121]. Interestingly, users tend to give credibility to another user's post not based on how soon they receive the information but on how many times they hear it from different sources (Flow) [125]. Influence analysis can vary from social network to social network as different parameters are used to measure it, e.g. follower and retweet count on Twitter or friends and likes on Facebook.

However, a general approach requires a centrality analysis of the network as it identifies the most important actors in it attending to multiple parameters [17] [121], [126] **degree centrality**, which measures the number of links held by each node, helping find connected and popular ones; **betweenness centrality**, which measures the number of times a node is on the shortest path between nodes, and helps find nodes which influence the flow of information in the network; closeness centrality, which measures how close is a node to all other nodes of the network, helping find influential and well placed nodes; and finally, **eigen centrality**, similar to degree centrality but taking into account how many links the connected nodes of a given one are, giving a more general analysis of a node.

Given the massive scale of social network graphs, some of these calculations may be too computationally intensive to perform on the whole network. For this reason, several alternative algorithms are available:

- *Effective Closeness* measures an approximation of the number of neighbour nodes in a given number of steps, much more efficient than standard closeness [127].
- *LineRank* is an alternative to the betweenness measure, vastly more efficient measuring the flow through a node without explicitly constructing the line graph [127].
- *PageRank* and *HITS* algorithms, originally proposed to rank web pages by authority, can also be used for influence analysis [128].

3.4.4 Link Analysis

Link analysis is the evaluation of the connections between nodes based on repeated observation of those nodes, its interactions and relationships [29]. This data can be used to rank users by popularity or influence, detect communities, classify them in a given set of categories and predict the formation of future links between them [129]. Common approaches to link-based user ranking in social networks can be consulted in 3.4.3 and more modern methods, like Google's *RankBrain* algorithm, also use deep learning techniques in order to accomplish this task [29]. Likewise, community detection approaches can be found in 3.4.2.

[17] https://cambridge-intelligence.com/keylines-faqs-social-network-analysis/

Link-based classification is based on the fact that related users tend to share class labels. Several approaches have been proposed: extended machine learning algorithms with features that measure class label distribution in the Markov blanket of the user, computer vision algorithms like *relaxation labelling* and part-of-speech tagging as part of the natural language processing approach [129].

Link prediction tasks can be approached by both supervised and unsupervised methods, but current research recommends the use of supervised ones, which can outperform unsupervised methods by a good margin but require past knowledge of the analysed network [130]. Geodesic distance, or number of edges between two users, has proven to be one of the most important features for this task [29]. A general framework for a supervised approach to link prediction can be found in [130].

3.4.5 Multimedia Analysis

Multimedia analysis is the automatic information retrieval in a natural language form media available on a social network. Images are the most common type of media available on social networks, so the focus should be on analysing them. Due to the nature of social media, the logical approach to image information retrieval is *Content Based Information Retrieval*, based on the features of the pictures themselves as they usually lack human assigned labels. Feature extraction is the basis of this technology, extracting both primitive features (colour, texture or shape) and domain specific features (i.e. human faces or finger prints). *Colour-Histogram* is the most common method for colour feature extraction, shape extraction is usually approached by *Fourier descriptors* and *Moment invariants* and texture extraction can be approached by structural methods like *morphological operator* and *adjacency graph*, and statistical methods like *Fourier power spectra, co-occurrence matrices* or *fractal model* [131]**.** Domain specific feature extraction usually involves supervised machine learning algorithms with labelled sets of data, but some unsupervised approaches have been proposed like the one in [132]**.**

3.4.6 Trend Analysis and Topic Detection Tracking

Topic detection tracking tries to automatically answer the question of "What, when, where and by whom are the popular topics/events/trends set". This task makes use of machine learning methods, be it unsupervised for unspecified event analysis or supervised for the detection of specific events, with two main approaches [29]**,** [133]:

- ***Feature Pivot*** techniques monitors topics previously unseen or rapidly growing, generally focusing on change and burst detection of text features, defining bursts as term or sentiment deviations. Several approaches have been proposed, like using sentiment analysis to measure user response to an event, detecting peak of hashtag usage using the *Discrete Wavelet Transformation* or making use of geo-localization to detect local events.
- ***Document Pivot*** techniques cluster documents according to their textual similarity, usually labelling them as "Event-Related" or "Not-Event-Related". These methods make use of supervised or unsupervised machine learning, employing *Naive Bayes* or *Support Vector Machine* for the former and a *scoring function* for the latter. *Document Pivot* however has several limitations, as the techniques aren't directly applicable to social media like Twitter and require batch processing, so they are not scalable to large amounts of data. Both *Feature Pivot* and *Document Pivot* can be combined into hybrid methods.

There is another unsupervised learning technique alternative, which is to model normal user behaviour and detect deviations from it in a manner quite similar to *Feature-Pivot* techniques. It's also been found that Twitter is the first place for topics to emerge thanks to its status as an information and social network [29].

It's important to know that trend analysis and topic detection tracking is never an easy task, as it implies real time analysis of vast amounts of data and a great deal of noise, especially in general purpose social networks like Twitter and Facebook [133].

3.4.7 Collaborative Recommendations

Recommender systems' (see Section 4.2) accuracy can be greatly improved by analysing the data provided by social network users. To correctly analyse and incorporate this data to recommender systems two main approaches are proposed [29], [134], [135]: ***Content-based*** methods analyse documents, and descriptions of items rated by a user and construct a model of the user's interests based on the features of said items and comparing these interests to the features of other items, retrieving the level of interest. This approach has several advantages, as the recommendations depend on just the active user, explanations of how the recommender system works can be provided and are able to recommend items not yet rated by any user. It has some drawbacks too, like an over-specialized system that only provides items of the classes in the user's interests and the impossibility to provide relevant recommendations to new users of a social network. ***Topology-based*** or ***collaborative filtering (CF)*** approaches are the most common and base recommendations on the behaviours of other users in the system, intuitively assuming that if users agree about the relevance of some items, they'll likely agree about other items, even if some of these users haven't yet seen them. Most algorithms work by first modelling user's preference and then producing recommendations by ranking candidate items. The algorithms used in *collaborative filtering* are *User-User Collaborative Filtering*, which finds other users with similar past rating behaviour and use their ratings to provide recommendations to the active user, and *Item-Item Collaborative Filtering*, which uses similarities in the rating patterns of items instead of users, making it similar to content-based methods but with the information being deduced from preference patterns instead of item data. Hybrid approaches of these two methods are also explored in [135] and their efficiency varies on a per-application basis.

Recommender systems face several issues, like data scarcity, the cold star problem and user privacy concerns, even more so in the context of social networks. The cold star problem refers to the limited amount of available data from a new social network member, which limits the accuracy of the recommendations. User privacy is being addressed by *Privacy-preserving collaborative filtering* using various techniques and algorithms to protect individual privacy during recommendation [29].

4 State of the art on PAPUD use cases

4.1 Improve Customer Experience in E-commerce

The Web allows people to share information from large database repositories. The amount and heterogeneity of information require a hybrid solution based on natural language processing, information retrieval, and semantic Web. Indeed, there are many search engines available today. Most of these search engines use keywords for query answering. Search engines usually search and return a set of web pages that match user query. However, it has been shown that such search engines fail when answering intelligent queries that require some reasoning. They either show inaccurate results with this approach or show accurate but unreliable result. In this section, we will make a preliminary survey over the existing literature regarding intelligent semantic search engine [115].[2]

XSearch is one of the most known smart engine [136] based on semantic search. With its simple query language, it constitutes a suitable tool for native users. XSearch is a Web-based search application that was designed to accomplish two main functions: search integration and search result contextualization. The system permits users to search across multiple databases and display integrated results in a single consolidated listing. Moreover, the results are annotated to indicate whether there are associated contextual cross-references. Advanced indexing techniques built on top of Tf-Idf techniques were used for the development process.

SEWISE is an ontology-based Web information system proposed in [137]. It supports Web information description and retrieval based on ontology reasoning. SEWISE can map text information from various Web sources into one uniform XML structure and make hidden semantic in text accessible to program. The textual information is extracted by Web Wrappers from heterogeneous. Text mining techniques such as categorization and summarization are used to process retrieved text information for query answering.

The Algolia model provides search as a service, offering web search across a client's website using an externally hosted search engine[18]. Although in-site search has long been available from general web search providers such as Google, this is typically done as a subset of general web searching. The search engine crawls or spiders the web at large, including the client site, and then offers search features restricted to only that target site. This is a large and complex task, available only to large organizations at the scale of Google or Microsoft.

Algolia's product only indexes their clients' sites and so the search task is far simpler. Data for the client site is pushed from the client to Algolia via a RESTful JSON API then the search box is added simply to the client's web pages. This search model is intended to give the performance and sophistication advantages of a full in-house search engine operating on the native web site back-end database, but with the simplicity of setup of using a site-restricted Google search.

Elasticsearch is a search engine based on Lucene. It provides a distributed, multitenant-capable full-text search engine with an HTTP web interface and schema-free JSON documents. Elasticsearch is

[18] https://www.algolia.com/

developed in Java and is released as open source under the terms of the Apache License. Official clients are available in Java, .NET (C#), PHP, Python, Apache Groovy, Ruby and many other languages[19]. According to the DB-Engines ranking, Elasticsearch is the most popular enterprise search engine followed by Apache Solr, also based on Lucene.

Elasticsearch is developed alongside a data-collection and log-parsing engine called Logstash, and an analytics and visualisation platform called Kibana. The three products are designed for use as an integrated solution, referred to as the "Elastic Stack" (formerly the "ELK stack").

Semantic Web Search Engine (SWSE) [138] is a smart engine which consists of crawling, data enhancing, indexing and a user interface for search, browsing and retrieval of information; unlike traditional search engines, SWSE operates over RDF Web data - loosely also known as Linked Data - which implies unique challenges for the system design, architecture, algorithms, implementation and user interface. SWES has successfully combined Semantic Web technologies with textual search. The search result is improved using both reasoning and external resources such as Wordnet.

4.2 Recommendation Engines

A **recommendation engine** (or system) is an algorithm that utilizes various sources of data and analyzes the past user behavior to suggest items which they are most likely to prefer. A recommendation system uses data analysis techniques to discover patterns in the data set and figure out the items that match the users' needs and interest. The ultimate aim of any recommendation engine is to stimulate demand and engage users. As sources of data, the recommendation systems uses: user preferences, item features, past user-item interactions and other additional information such as temporal and spatial data.

Traditionally there are three types of recommendation models: collaborative filtering, content based filtering, hybrid models. Content based systems use only the user-item interaction and do recommendations based on a similarity comparison between the content of the items and a user's profile. The feature of items are mapped with the feature of users in order to obtain user – item similarity. The top matched pairs are given as recommendations. Collaborative Filtering algorithm uses user preferences and item preferences such as transaction history, ratings, selection and purchase information. Other users behavior and preferences over the items are used to recommend items to the new users. In this case, features of the items are not known. Hybrid models uses both of those interactions and adds the user and item meta data [139, 140]. These models face two main problems, data sparsity and cold start problem. In the first problem, there are too many undesirable products compared to the user's preferred products and the entire product range. For example, at present, there are 500 million products in the Amazon system. If a user has even bought 1000 items, there will be a lot of untapped items left. This causes the generated vectors to contain 0 values in abundance.

Modern recommendation engines integrate deep neural networks to the approaches explained above. Below we describe a berief review of the current state of the art [141]. We will introduce these works in two main categories, recommendation systems using single deep learning technique and deep

[19] https://www.elastic.co/

composite model (recommender system which involves two or more deep learning techniques) and also describe whether the model integrates to the traditional methods.

DEEP LEARNING BASED RECOMMENDER SYSTEM:

1) Multilayer Perceptron based Recommender System: It has more than one layer and works with a feed-forward calculation. It is one of the oldest methods used in the literature. Both the increase in the front and high speed computers have made it possible to build such models. Suggestions are made from the similarities between users and products in content based models. The creation of a two-way network of links between users and products is commonly used.
    i) *Neural collaborative filtering* captures non-linear relationship between users and items and uses deep convolutional neural networks items to produce user-item latent vectors [142]. Cross-domain Content-boosted Collaborative Filtering Neural Network adds the content such as user and product information to the interaction matrix [143].
    ii) *Wide & Deep Learning* initially introduced in Google play for App recommendations [144]. It consists of a single layer perceptron (wide learning) and a multilayer perceptron (deep learning). The recommendation system can capture both memorization (features from historical data) and generalization.
    iii) *Deep Factorization Machine* integrates factorization machine and MLP [145]. It uses factorization machine for low-order interactions between features by means of addition and inner product operations, and deep neural networks for high-order, non-linear interactions.
    iv) *Alternative collaborative filtering* is a two-level attention mechanism (item-level and component-level) to integrated to a traditional collaborative filtering model.

2) Autoencoder based Recommender System: This model is based on unsupervised learning. Using model inputs and hidden layers, it tries to guess the similar inputs. It prevents the noise while estimating its input data. It is especially used during improvements on the data.
    i) *AutoRec* [146] takes users and items as partial vector inputs and reconstructs in the output layer. It has two different approaches item-based AutoRec, which usually performs better and user-based AutoRec. The performance can be improved by adding more layers to the deep network, using different combinations of activation functions and expanding hidden layer capacity.
    ii) *Collaborative Filtering Neural Network (CFN)* [147] is an extension of AutoRec which uses noise reduction and user-item meta data.
    iii) *Autoencoder-based Collaborative Filtering (ACF)* [148]
    iv) *Collaborative Denoising Auto-Encoder (CDAE)* [149]
    v) *Collaborative Deep Learning (CDL)* [150] is a Bayesian deep learning model with a perception component (DL) and a task-specific component.
    vi) *Collaborative Deep Ranking (CDR)* [151]

3) Convolutional Neural Network based Recommender System: It is a model in which, feedforward convolutional layers are present and the pooling operations are applied. The most important

aspect of this model is that it can include both global and local factors. It performs especially well in topologies with grid features such as image and area.

i) *Attention based CNN* [152] was proposed for hashtag recommendation. Words are encoded in a global channel of convolution filters. The attention layer is in the local attention channel with given window size and threshold for words.

ii) *Deep Cooperative Neural Network* (DeepCoNN) [153] is tightly coupled model that uses word embeddings technique combined with two coNNs.

iii) Loosely Coupled Models: Some of the models that loosely combine traditional methods with deep learning are, CNN for Image Feature Extraction, CNN for Audio Feature Extraction, CNN for Text Feature Extraction [154] [155] [156]

4) Recurrent Neural Network based Recommender System: This type of deep learning algorithms are successful particularly in modeling serial data. Unlike feedforward artificial neural networks, layers have loops and memory. The purpose here is remembering the calculation of the previous entry. The previous input state is stored and used with current the input data. It works in the serial data, such as text or voice, because it includes the calculations up to that moment. In traditional neural networks and machine learning algorithms, inputs are evaluated independently from each other. An entry is completely independent of the previous entry. The RNN algorithm is a solution to the vanishing gradient problem especially in feedforward networks. In this problem, the initial layers learn slower than the latter.

i) *Session-based Recommendation with RNN* [157] works particularly well on temporary and suddenly changing occasions, and with a set of features such as web click-stream.

ii) *Recurrent Recommender Network (RRN)* [158] takes into consideration the preferences that change with time and seasonal effects.

iii) *Neural Survival Recommender* [159]

iv) *Attention based RNN* [160]

v) *GRU Multitask Learning* [161]

5) Deep Semantic Similarity Model based Recommender System: It is a model used to learn semantic representations of entities. These models that are based on artificial neural networks are widely used in the field of information acquisition. It is of great importance performing such semantic calculations for the extraction of relations between concepts. This model is especially suitable for top-n advice. The DSSM model can express the users and products in a lower dimensional space using their properties. Thus, the relationship or similarity between items can be determined.

i) *Deep Semantic Similarity based Personalized Recommendation (DSPR)* [162]

ii) *Multi-View Deep Neural Network (MV-DNN)* [163]

6) Restricted Boltzmann Machine based Recommender System: The RBM is an artificial neural network that consists of two hidden layers that appear as one. There is no layer between these two layers. This model is the first deep learning model to be applied to recommendation systems. In this architecture, which requires binary structure as input, a large amount of data needs to be converted into a binary structure with one-hot encoding.

i) *Restricted Boltzmann Machine Collaborative Filtering (RBM-CF)* [164]

ii) *Hybrid RBM-CF* [165]

7) Emerging Methods: NADE and GAN

i) Neural AutoRegressive based Recommender System [166]

ii) Generative Adversarial Network based Recommender System. IRGAN [167]

DEEP COMPOSITE MODELS FOR RECOMMENDATION:

Here we list some of the deep composite models, that are combination of two models: CNN and Autoencoder [168], CNN and RNN [169], CNN and MLP [170], RNN and Autoencoder [171], RNN and MLP [172], CNN and DSSM [173], RNN and DSSM [174].

4.3 Call Center Operations

Call centers are modern service networks in which agents provide information or guideline services to customers via telephones [175]. They assume a centralized office used for receiving or transmitting a large volume of requests by phone and have become a key contact solution between service providers and their customers. Inbound call centers process calls that are initiated by customers. Examples include phone operations that support booking and sale for hotels, airlines, and car rental companies, as well as banks service centers, mobile telephony providers, brokerage firms. Outbound call centers initiate calls to outside parties, for example, to organizations such as marketing research and polling companies. A contact center is a location for management of individual communications, including letters, faxes, social media, instant message, e-mail, etc[20].

A simplified scenario for a typical inbound call center is the following: A customer calls the telephone number associated with the call center of the envisioned service; once connected, the customer usually first accesses an interactive voice response unit (IVR) and identifies himself or herself. In the IVR, the customer can perform some self-service transactions, and sometimes conclude the service there. In case she/he still wants to speak with an agent, if an agent is available and is able of performing the desired service, then the customer is redirected to the agent to receive service immediately; otherwise, the customer joins an invisible queue of waiting callers.

A brief overview of the main technologies involved that are used at a call center includes[21]:

1. Automatic call distributor: manages incoming calls.
2. Predictive dialer: automatically calls a list of telephone numbers, screening out busy signals, no-answers and answering machines, predicting at the same time when a call center agent will be available to handle a call.
3. Issue tracking system: records and manages the progress of customer-related issues.
4. Knowledge base: gives call center agents and customers easy access to a plenty of information.
5. Specific analytics such as: average speed of answer (ASA) or abandoning customer's rate.
6. Session initiation protocol (SIP): used in call centers in conjunction with Internet Protocol technology.
7. Media Resource Control Protocol (MRCP): represents a standard fir the developers of applications based on telephony and voice-based technologies.
8. Email response management software: automatically handles emails according to user-defined rules.
9. Text analytics: used to analyze text from customer service interactions.
10. Speech analytics: used to screen customer interactions to identify customer needs.

[20] https://en.wikipedia.org/wiki/Call_centre#cite_note-EWA-1
[21] https://www.quora.com/What-technologies-are-used-at-call-centers

According to this site[22] there are some compulsory solutions that are adopted at most of the state of the art performant centers. Among them:

1. Real-time speech analytics used to calibrate to customer needs and emotions, and thus to improve agent training, track compliance, in order to improve performance of the organization22.
2. Text Analytics
3. Chatbots: 2017 might have been the year where more call centers adopted chatbots to handle a good deal of basic calls and queries. They will not replace agents, but they should lighten an agent's workload so he or she can focus on the more difficult or complex customer issues

We further stress next one of the most challenging topic for call centers.

- **Speech Analytics**

Speech analytics software automatically processes and identifies important keywords and phrases, allowing fast retrieval of calls related to specific issues and challenges. It may also provide automated alerts when a new relevant call is received.

Nowadays, one of the important tendencies in software applications is including voice-based interfaces to facilitate the communication with the users and improve the quality of the provided services. This trend is emphasized in the 2017 Report of VoiceLabs [176] which points out the fact that at present there are already in use tens of millions of devices operating voice-based interfaces as the main mean of interfacing. These devices are operated on platforms that include software modules of speech recognition, speaker recognition and speech synthesis. Among the companies that provide such programs are Amazon (Echo, Amazon Tap, Echo Dot), Google (Google Home), Apple, Samsung and Microsoft.

An important role of voice based devices which use voice as the main means of interaction [177] (voice first) is that of intelligent assistant for the users. In the competition for voice banking the leaders in implementing voice assisted solutions are the companies Nuance and Personetics [177].

Nuance was used to implement the agent voice first in the framework of several call center applications of USAA, Santander (in USA, Mexico), ING (Holland, Romania), TATRA Banka, Tangerine Bank, SK Telekom. Personetics has facilitated implementation of the virtual agent Ally Assist for the banking application Ally.

At ING, in 2015, more than 100.000 clients already used in 2015 on their mobile banking applications a virtual agent based on voice called Inge, implemented with Nuance, for such operations as card balance checking, or payments operated by card. In 2015, the agent was supplemented with a biometric system based on voice print and/or finger print, to allow operations that require authentication (the PIN code or the password are replaced or doubled by the client's biometric prints). Thus, Inge, the ING version of Nuance, provides an interface contained by voice that eases the communication and also provides an increased level of security.

[22] https://www.monetsoftware.com/blog/workforce-management/post/five-technology-solutions-found-at-state-of-the-art-call-centers

Santander UK also has launched in 2016, a similar assistant based on voice for their mobile banking application in the frame of a package destined to students, also by applying the product of Nuance. For the moment the use of voice is limited to specific banking operations concerning information clients' transactions or card data. For the future development of some other services are considered such as biometric authentication.

Concerning the banking applications, the first payments products to use vocal interfaces Siri P2P were PayPal, Venmo and Square Cash. Some banks such as Monzo in England, the German bank N26, or Royal Bank of Canada (RBC) already operate these payment solutions P2P [177]. The analysis of the State of the art shows that there is an increased appetite for communication through automated voice interfaces operated in call centers. However, these applications are limited to a restricted set of operations. It is estimated that in the next 10 years 50% of the banking operations for instance will be accomplished with the help of Voice First devices.

In Europe among the organizations involved in projects related to optimization of the activities of call centers, speech processing and analytics and their use in voice interfacing solutions for communication with the clients are Genesys [178], Avaya [179], Vocapla[23], Calabrio[24], Verint[25]. In Romania such a call-center platform is operated by the ING bank, it was initially used on telephone devices installed inside the ING agencies, and at present is operated on mobile phones. This system allows the authentication through a personal identification code, sometimes coupled with biometric authentication and pick a specific service among a range such as card blocking, card balance checking, adapted to the Romanian language and bases on a limited vocabulary.

Another example is the call center operated by Vodafone Romania and using voice interfaces, but with a very bad average speed of answer rate. A formal evaluation of the performances of these applications is not available but empirical evaluations indicate a success rate of about. A classical diagram describing a common call center system is presented in Figure 12 below:

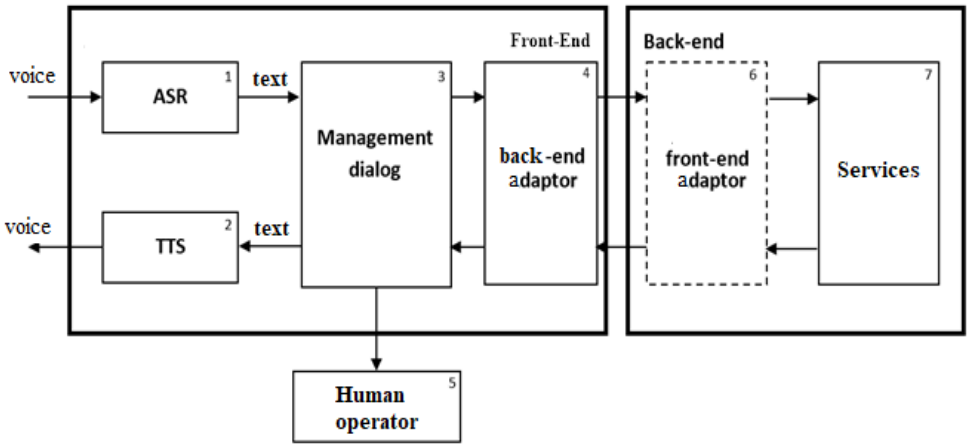

Figure 12 - General diagram of a voice-based call center system

[23] http://www.vocapia.com/
[24] http://calabrio.com/
[25] http://www.verint.com/

The interesting components in the above scheme are:

- ASR (Automatic speech recognition) – is responsible for translating the speech signal into text
- TTS (Text to Speech) accomplishes a voice synthesis function translating a text into a speech signal.

The Speech recognition function is implemented by the organizations enumerated above using specially designated products. Although at present the implemented systems usually implement

- the MFFC technique in the feature extraction step
- HMM in the classification and recognition step

The state of the art in Automatic Speech recognition is considered to be based on Deep Neural Networks [26] approach, which is already implemented by several open-source software tools such as KALDI, Nuance, PoketSphinx, for possible use in voice-based applications. . An active area of research like this is difficult for such toolkits to support well, because the state of the art changes constantly therefore code changes are required to be updated, and architectural decisions may need to be reconsidered [27]. For instance Kaldi currently keeps three different codebases for deep neural networks. The first one, maintained by Karel Vesely, allows simpler implementations and is easy to modify. The second is maintained by Daniel Povey and is more flexible as it supports multiple GPUs. The third is intended to be the recommended path going forward. All are still active in the sense that the up-to-date recipes refer to all of them. This toolkit also enables implementation of hybrid solutions such as DNN-HMM to speech recognition issues. F or instance, at Beia a voice-based application intended for a call center was devised using KALDI[28].

All important names in this field, such as IBM, Microsoft, Apple, Amazon or Google have already adopted this trend in their ASR products. A more up to date diagram of a call center is presented in the schema below. Along with the real-time automatic speech recognition it considers other functions to be accomplished such as:

1. Automatic speaker recognition
2. Emotion detection

[26] http://neuralnetworksanddeeplearning.com/
[27] http://kaldi-asr.org/doc/about.html
[28] http://kaldi-asr.org/doc/about.html

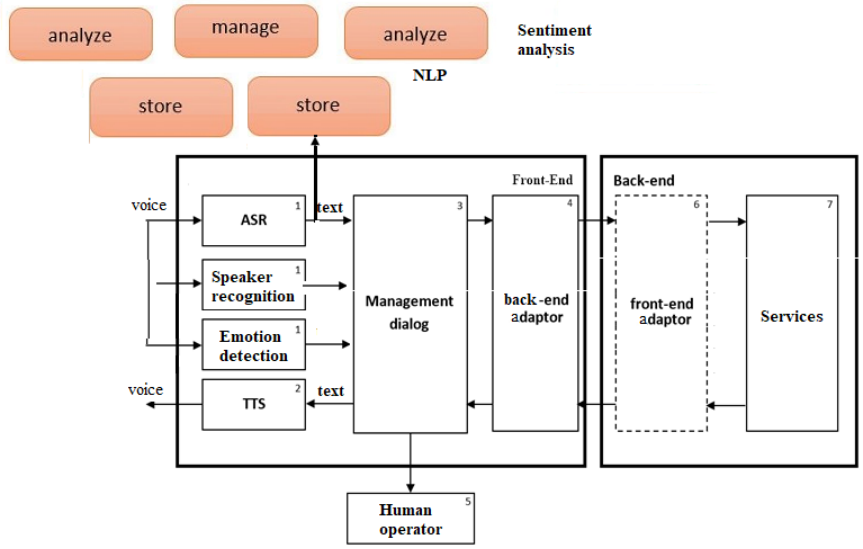

Figure 13 - General diagram of a voice-based call center system and the underlying Big Data processes and analytics

The speech data, the translated and aligned text is usually stored, and the new information may be inferred using analysis tools such as NLP to provide for instance Sentiment analysis. Beyond speech recognition, Deep Neural Networks are also often applied in speaker recognition [180] [181] and emotion detection, as seen previously in this document [182] [183].

4.4 Human Resources (HR)

An effective e-recruiting engine can help job seekers easily access recruitment opportunities and reduce the recruitment burden by providing suitable items that match their personal interests and qualifications. It also frees companies from information overload and advertisement cost.

Sharing personal information is a common activity among social networks users, which allows and leads employers to use social networking sites to screen job applicants. There are several reasons for this: First, social networks allow employers to verify information provided by the candidates, as ADP Screening and Selection Services found that applicants do falsify information [184]. Second, social networks are also easily accessible and usually free, which vastly broadens the scope of the search. Finally, employers also use these services to search for negative information in order to avoid negligent hiring, as they can be legally responsible for employees who cause harm to the third party. Social networks provide the means to acquire as much information as possible about the candidate, which is in the best interest of the employer, but the volume of data available per applicant and the amount of social network users makes it quite laborious, if not impossible, to filter it manually [29].

An automated approach becomes almost necessary, but a few considerations need to be taken into account. Privacy concerns are to be expected, as there are several barriers (legal, sociopolitical or psychological) concerning the disclosure, mining and processing of personal information. Also, the inference of social characteristics based on data extracted from social networks is still a fairly new and open research venue, yet to mature, and may not be completely accurate [185].

Recruiting programs use different AI technologies like:

- **Recommender systems**. Many researchers have been conducted to discuss different recommender system related to the recruiting problem [186]. Among them, [187]

discussed a bilateral matching recommendation systems to bring people together with jobs using an Expectation Maximization (EM) algorithm, while Golec and Kahya [188] delineated a fuzzy model for competency based employee evaluation and selection with fuzzy rules. Paparrizos et al. [189] used Decision Table/Naive Bayes (DTNB) as hybrid classifier. These systems used many manual attributes and various information retrieval techniques. The results can be improved by using deep learning methods to accelerate the process of finding the most appropriate jobs. Hence Zhang et al. [190], try to optimize the qualified worker-position matching, considering various characteristics of workers.

- **Text Mining**. Structuring, extraction and integration of CVs information are important activities in the recruitment process. Indeed, the information retrieval in CVs, depends on content modeling, which aims to assign a semantic to improve the quality of retrieval results. Text mining tries to solve the information overload problem by using techniques already reviewed in this document, such as machine learning, Natural Language Processing (NLP), Information Retrieval (IR), and knowledge management [191]. Berio et.al. [192] was interested on applying knowledge techniques to competence extraction, and management. He used ontology techniques, e-learning system and CRAI model (Competency Resource Aspect Individual) to provide a representation of competencies required and acquired. [193] outlines a HR-XML based prototype dedicated to the job search function. The system selects important information such as pay-check, topic, abilities, etc. Moreover, another approach in e-recruitment domain [194]used Finite State Transducer formalism to extract key information from CV. This work proposed a model for the representation of the CV's content, and used Finite State Transducer formalism for this representation. So, the parser browses the CV in the XML format and identifies the different tags which should be in relation with employer requirements to construct the transducer, ensuring an easy and efficient CV retrieval.
- **People analytics** is the use of data and data analysis techniques to understand, improve and optimize the people side of business. People analytics links people data (inputs) with different types of business data using predictive algorithms to produce outcomes (outputs) aligned with company goals such as increased revenues and lowered costs.
- **Predictive analytics** can be applied to candidates to predict which ones are likely to be successful employees. This predictive model can be created using resume data, pre-hire assessments, or interview scores. For a predictive model that uses resume data as its inputs, the multiple variables could include education level, years of experience, skills, and personality traits.
- **Sentiment analysis**. Applicant personality traits are considered critical in most job positions, but are overlooked in existing e-recruitment systems. Previous work indicates that blogs contain textual features reflecting the author's personality [195] as well as his mood and emotions [196]. Specifically, it has been established that extrovert people use many social words and positive emotion words, and few negative emotion words. It has been shown in [197] that there are significant correlations between word categories and the corresponding personality traits.

There exist some companies with AI based Recruiting products, for example:

Ideal (***https://ideal.com/product/***) uses Artificial Intelligence for Recruiting. The products of the company are:

1. Candidate Screening - Automated Resume Screening Software Using AI, in real-time. Ideal's software goes beyond simple keyword matches identifying the most qualified candidates, instantly. It scans and filters resumes grading every candidate A, B, C or D; uses

artificial intelligence to scour external and internal databases 24/7. Ideal says: "our customers have been able to reduce time to fill between 25% and 75%! Reach out to the best candidates in days, not weeks". Ideal uses the feedback of the recruiting companies to get smarter and smarter.
Candidate sourcing - Ideal's AI can source candidates from external databases or from already existing company's database.

2. **Recruiter Chatbot**
   Ideal's chatbot can reach out to candidates to further qualify them, while keeping them informed and engaged. The Chatbot asks custom questions such as, "Can you lift 50 kg?" and "When are you available to start work?"
3. **Candidate Rediscovery**
   Ideal's technology can rediscover past candidates from your database that are a great fit for a new role

X.ai (https://x.ai/). AI scheduled meetings for smarter work days; Once you connect your calendar and update your preferences, company's AI assistants will schedule your meetings with clients, contacts, leads and candidates. 24/7.

Career Spark ***(https://www.careerspark.com/)*** saves recruiters sourcing time by automatically finding and ranking candidates. Success-based matching technology leverages science and objective data to help leaders understand what is driving success so they can use that information

1. Predict which of your candidates has highest potential to succeed in a role, based on advanced science and the data. No more guesswork, just educated, informed hiring decisions.
2. Career Spark will provide you with clarity on which growth opportunities in your company are most aligned with each of your employees' potential.
3. Building and managing diverse teams: Our system intelligently converts data into insights for superior feedback, coaching, and development – to help you get the most from your team.

Harver (https://harver.com/). Harver is a pre-hiring and talent matching platform, designed to make excellent hiring achievable for companies of all sizes. Choose from a wide variety of scientifically proven pre-employment assessments that are easy to setup and customize to your hiring needs. AI matching algorithm for true data driven hiring.
Leverage the power of AI through our proprietary self-learning matching algorithm that gets better over time. With every candidate that passes through Harver, the platform optimizes the factors that are predictive for quality for hire.

Also there are services that can be used for solving AI based Recruiting products as well Call Center products:

Microsoft Azure:

1. For search: - Cognitive Services: Bing Web Search; Bing Video Search; Bing Image Search;
2. For Data analytics: Data Lake Analytics; Bot Services
3. For Sentiment analysis: Cognitive Services: Text Analytics

4.5 Behavior Analysis for Reverse Efficient Modeling (BAREM)

By analyzing user's activity in an eService, BAREM aims to detect navigation problem or misusing of the eService and to notify the designer with regard to the level of frustration resented by the users.

In order to achieve this purpose several issues need to be addressed:

- Log file analysis – reconstruct navigation sessions and extract recurring patterns
- Sentiment analysis – quantifying the frustration level resented by the users using non-intrusive and privacy preserving sensors

4.5.1 User log file analysis

There is a lot of literature and tools providing log file analysis for user activity on web sites[29]. Most of them propose to aggregate different log source and offer dashboard presentations with features to allow the user to search for specific patterns in log records[30] [31].

Usually, the main purpose of these tools are Search Engine Optimisation, increasing website frequentation based on analytical tools using log files. In the BAREM use case, this type of analysis is irrelevant because they focus their analysis on the number of unique user visiting a webpage. In the BAREM use case we are interested in the navigation pattern of different users.

Some efforts were done by academic community to analyse log files of webservers to extract user's behaviour[32] but the goal of this type of data mining is to find system errors or broken links like what we discussed in section 3.3.2, not a deep analysis of navigation scenario.

4.5.2 Sentiment Analysis from video and user navigation

The latter part is subject to user acceptance and specific attention needs to be paid in order to fulfill the sentiment analysis task while reducing privacy intrusion. In order to achieve this goal, we foresee inferring sentiment analysis from navigation analysis. Navigation analysis is basically an objective measure. At the beginning, In order to acquire sentiment information other sensors such as video and biosensors are used. Co-occurrence analysis of navigation sessions and sentiment analysis is done in order to infer in latter stages the sentiment from navigation path only. In the following we briefly introduce the state-of-the-art related to the above activities.

Sentiment analysis is one of the most important research area in many fields. Since the 1970s and the work of Paul Ekman who emphasized the correlation between emotions and facial expressions [198], works on emotion recognition have blown up especially in computer vision. In the literature, extracted features dedicated to facial expression recognition can be divided into two kinds: "handcrafted" (or predefined) features and learned features.

On one hand, "handcrafted" features are the first characteristics to have been defined in this context. Most of the time, they describe the appearance, the geometry, or both, of faces. Among appearance descriptors, one can cite Local Binary Patterns (LBP) [199], Gabor filters [200], *Local* Phase Quantization (LPQ) [201] which permits to obtain relevant features on a single frame. Concerning geometry features, landmark locations [202], geometric distance [203], depth maps [204] and so on can also be used. These two kinds of features can be merged to improve the performance [205]. To

[29] http://loggly.com
[30] http://www.splunk.com/web_assets/pdfs/secure/Splunk_for_Application_Management.pdf
[31] http://www.loglogic.com/products/log-management/log-management-infrastructure
[32] https://pdfs.semanticscholar.org/3932/09e5dabf41b02ac9352a2cfa5dc12878789f.pdf

take into account that facial expression is a spatio-temporal phenomenon, dynamic appearance descriptors and dynamic geometric descriptors have been defined as LBP-TOP or LPQ-TOP [206], MHI [207], LMP [208].

On the other hand, recent learned features are defined thanks to so-called deep learning methods. In this case, features are not defined directly by hand but they are learned from a big dataset using deep neural networks. In this context, one can cite Convolutional Neural Networks (CNN) [209], AU-aware Deep Networks (AUDN) [210], deep Boltzmann Machine (DBM) [211].

In some cases, it is difficult to recognize an emotion using a single modality (only video, audio or some biological sensor). Indeed, if the person hides his face or does not speak, extracted features from the different sensors taken individually could not be enough to extract his emotional state. To recover as much information as possible from the sensors used, multimodal fusion methods have been proposed. The most direct way to merge the characteristics of several modalities is to concatenate the descriptors to obtain a unique feature vector [212]. In this case, the challenge consists in merging features of different natures, which is not always obvious. It is called the early fusion. Recently, in [213], multiple kernel learning (MKL) is used to merge visual, audio and textual data for multimodal emotion recognition. The second kind of approaches is called late fusion. In this case, decisions are merged instead of the features as in [214]. More recently, deep learning methods are used to merge multi-modal features. For example, multiple-fusion-layer based ensemble classifier of stacked autoencoder (MESAE) [215] is used to merge multimodal physiological signals. In [216], two CNNs are computed on color and depth images and the outputs are merged using a fully connected layer in order to perform RGB-D object recognition.

4.6 Prescriptive Maintenance for HPC

There are two main fields according to the types of data considered: system logs analysis and metrics analysis.

4.6.1 Logs analysis

To predict failures based on logs, the first step is to turn textual data into numerical data. In fact, machine learning algorithms can only work with numerical values. After this step two main ideas are used:

1. Identify the normal behavior and detect anomalies when the behavior diverges.

2. Predict errors based on the previous observed errors.

4.6.2 Logs transformation

To turn textual data into numerical data, several techniques can be used. Most of the techniques are provided by Natural Language Processing (NLP) [217] approaches. One of them is based on detecting variable parts in logs and using these parts to create vectors of features [218]. These features can be, for example, the type of the log entry, and the value of variables. Another technique is based on a dictionary [219]. The idea is to parse ”MESSAGE” field to take out numbers, punctuations, and file names then build a dictionary of phrases and associate one number to each phrase. We provide an example bellow.

• A - 15/12/2006 15h21m30s Directory missing: /home/folder

• B - 15/12/2006 15h42m12s Directory missing: /root/folder temp

• C - 15/12/2006 15h50m06s Connection of user.

In this example, logs A and B are the same, because the filenames are removed during the filtering step. Thus, the dictionary has only 2 entries:

1. - Directory missing

2. - Connection of User

Logs can be replaced by dictionary indexes:

• A - 15/12/2006 15h21m30s 1

• B - 15/12/2006 15h42m12s 1

• C - 15/12/2006 15h50m06s 2

After this step of transformation, prediction can be made.

4.6.3 Behavior analysis

Behavior analysis is based on normal behavior detection. ”Normal” behavior is defined as the most frequent behavior. For example the normal behavior can be the following sequence of logs Log A ==> Log B ==> Log C. If the next occurrence of Log B is followed by Log D, Log A ==> Log B ==> Log D, behavior analysis can detect the abnormal behavior because Log C is different from Log D. Xu et al. [220] detect unusual patterns based on Principal Component Analysis (PCA). Normal patterns are found by the PCA, outliers can be detected with a distance greater than a threshold. By using this algorithm, authors discovered a hidden bug in Hadoop Distributed File System (HDFS).

4.6.4 Anomaly analysis

Anomaly analysis is based on finding correlations between logs at time t and logs at time t-1. For example, the sequence of logs can be Log A => (after 2 min) Log B => (after 3 min) Log errors. In this example, when Log A appears, Log error will appear within the 5 next minutes. Hence, the prediction of the error can be done as soon as Log A appears. Liang et al. [219] propose an approach that uses multiple fields in logs (severity, timestamp, job id, location and message) to create vectors with 5 different features. After this step, machine learning algorithms (RIPPER, SVM and Bi-Modal Nearest Neighbor) are used for learning correlations. With logs of an IBM BlueGene/L machine, authors are able to predict failures using nearest neighbors between 6 and 12 hours in advance with a prediction accuracy more than 50%.

4.6.5 Metrics analysis

A metric is represented as a time series. A time series is a series of numerical points in a temporal order. The study of time series is divided into two main fields: classifying each time series according to various properties (time series classification [221]) and predicting the next value of a time series

according to the time series modelling (time series prediction [222] [223]). Work about time series classification is very large. Bagnall et al [221] present a useful survey for time series classification. The best algorithms with respect to complexity and precision are DTW (Dynamic Time Warping) for features extraction and Nearest Neighbor algorithm for classification. Multiple open datasets[33] containing representative time series pattern as increasing, decreasing, square (increasing, stable, decreasing) are used to compare algorithm performance.

4.6.6 Predictive maintenance using metrics

In the literature, we find several work that try to implement preventive maintenance using metrics for computing systems. Predictive maintenance can be made on several systems including hard drive [18], network [224], CPU [225], etc. Hard drives have several sensors (SMART sensors) for monitoring their states. Thus the failures prediction based on statistical analysis [226], or machine learning [227] can be done. Another approach is to try to optimize resource utilization. It can be done by creating new thermal scheduler [225] for example or optimizing network by TCP classification [224]. Detecting overheating based only on metrics is possible. Hsu et al. [228] propose method based on threshold. They predict the next point of temperature and if this next point reaches the threshold, an overheating is predicted.

[33] http://www.cs.ucr.edu/~eamonn/time_series_data/2018-04-18

5 Conclusion

This document surveys the researches and state of the art on deep learning techniques and their applications in different areas such as natural language processing, image, voice, and video processing, social networks and log files analysis, big data analytics, call centers and many cores machines. In addition, this deliverable dedicates two sections to general topics on deep learning and one section to the project use cases.

The state-of-the-art presented on deep learning distribution on many cores machines will help the project to propose novel performance evaluation techniques using different deep learning frameworks.

Data analytics and general deep learning methods presented in third section, providing a general perspective of descriptive, predictive and prescriptive analysis, techniques and tools of big data, help to the project use cases to get informed of existing NLP methods for sentiment analysis and relation extraction and proceed to improve or propose novel and more efficient models. Text analytics with applications on popularity detection and user log files and some deep learning techniques to use in online social network processing and analysis are introduced in third section, which will be exploited to develop new methods to enhance user experiences, e-commerce services efficiency, recommendation systems performance, and so on. The final section discusses deep learning techniques used in Papud use cases including:

- Improve customer experience and recommender systems efficiency in e-commerce
- Improve and propose new models for call center operations using speech analysis methods
- Improve job recommendations in human resources use case using text and sentiment analysis, and recommendation techniques
- Exploit user behavior analysis for reserve efficient modeling exploiting user log files and social networks data analysis
- Propose novel models for prescriptive maintenance for HPC

References

[1] J. Dean, C. Greg, M. Rajat, C. Kai, D. Matthieu, L. Quoc, M. Marc, R. Marc'Aurelio, S. Andrew, T. Paul, Y. Ke and N. Andrew, "Large Scale Distributed Deep Networks," in *NIPS 2012: Neural Information Processing Systems*, Lake Tahoe, Nevada, United States, 2012.

[2] D. Amodei, R. Anubhai, B. Eric, C. Carl, C. Jared, C. Bryan and e. all, "Deep Speech 2: End-to-End Speech Recognition in English and Mandarin," archiv, 2015.

[3] K. Alex, "One weird trick for parallelizing convolutional neural networks," google Inc, 2014.

[4] S. Gupta, A. Agrawa, K. Gopalakrishnan and P. Narayanan, "Deep Learning with Limited Numerical Precision," in *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, 2015.

[5] NVIDIA, "cuDNN," NVIDIA, 2016. [Online]. Available: https://developer.nvidia.com/cudnn. [Accessed 2016].

[6] Intel, "how-xeon-phi-processors-benefit-machine-learning-deep-learning-apps-and-frameworks," Intel, [Online]. Available: https://software.intel.com/en-us/blogs/2016/06/20/how-xeon-phi-processors-benefit-machine-learning-deep-learning-apps-and-frameworks#. [Accessed 2016].

[7] Jia, Yangqing, Shelhamer, Evan, Donahue, Jeff and Kar, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the ACM International Conference on Multimedia*, 2014.

[8] J. Bergstra, O. Breuleux, F. Bastien and P. Lamblin, "Theano: a CPU and GPU math expression compile," in *Proceedings of the Python for Scientific Computing Conference (SciPy)*, 2010.

[9] F. Seide, H. Fu, J Droppo, G Li and D Yu, "1-Bit Stochastic Gradient Descent and its Application to Data-Parallel Distributed Training of Speech DNNs.," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[10] K. Alex, "The CIFAR10 dataset," [Online]. Available: https://www.cs.toronto.edu/~kriz/cifar.html. [Accessed 2016].

[11] S. P. Y, S. D and P. J. C, "Best practices for convolutional neural networks applied to visual document analysis," in *Document Analysis and Recognition*, 2003.

[12] I. T. Union, "Ict facts and figures 2016," http://www.itu.int/en/ITUD/Statistics/Pages/facts/default.aspx, 2017.

[13] G. Blackett, "Analytics network – O.R. & analytics," https://www.theorsociety.com/Pages/SpecialInterest/AnalyticsNetwork_analytics.aspx., 2013.

[14] Datafloq, "How to Select the Right Prescriptive Analytics Technique for Your Business," https://datafloq.com/read/select-prescriptive-analytics-technique-business/1977, 2018.

[15] E. B. J. L. a. H. K. T. Hu, " Tales of Two Cities: Using Social Media to Understand Idiosyncratic Lifestyles in Distinctive Metropolitan Areas," *IEEE Transactions on Big Data,* vol. 3, pp. 55-66, 2017.

[16] D. P.-P. a. T. Cohn., " Mining User Behaviours: A Study of Check-in Patterns in Location Based Social Networks," in *Proc. of ACM WebSci,*, 2013.

[17] E. I. M. G. K. a. J. C. G. Vlahogianni, "Short-term traffic forecasting: Where we are and where we're going.," *Transportation Research Part C: Emerging Technologies,* vol. 43, pp. 3-19, 2014.

[18] D.-A. H. a. K. M. Kitani, "Action-Reaction: Forecasting the Dynamics of Human Interaction.," in *Proc. of ECCV*, 2014.

[19] S. S. N. L. a. C. M. A. Noulas, "Mining User Mobility Features for Next Place Prediction in Location-based Services.," in *Proc. of ICDM. IEEE*, 2012.

[20] Y. L. J. H. D. W. H. S. J. S. a. Y. L. F. Xu, "Big Data Driven Mobile Traffic Understanding and Forecasting: A Time Series Approach.," *IEEE Transactions on Services Computing,* vol. 9, p. 796–805, 2016.

[21] B. B. a. M. O. B. Ghosh, "Bayesian Time-Series Model for Short-Term Traffic Flow Forecasting.," *Journal of Transportation Engineering,* vol. 133, p. 180–189, 2007.

[22] S. H. M. B. Gröger C., "Prescriptive Analytics for Recommendation-Based Business Process Optimization.," *Business Information Processing,* vol. 176, 2014.

[23] S. K. D. H. M. K. J. J. D. L. S. J. H. a. S. W. Song, "Prescriptive analytics system for improving research power.," in *In Computational Science and Engineering (CSE)*, 2013.

[24] S. D. C. G. C. &. J. H.-A. Rusitschka, "Adaptive middleware for real-time prescriptive analytics in large scale power systems.," in *Proceedings of the industrial track of the 13th ACM/IFIP/USENIX* , 2013.

[25] InformationWeek, "Big Data Analytics: Descriptive Vs. Predictive Vs. Prescriptive,," https://www.informationweek.com/big-data/big-data-analytics/big-data-analytics-descriptive-vs-predictive-vs-prescriptive/d/d-id/1113279, 2018.

[26] J. a. H. L. Fan, "Statistical analysis of big data on pharmacogenomics," *Advanced drug delivery*

reviews, vol. 65, no. 7, pp. 987-1000., 2013.

[27] Ontotext, "what is Information Extraction? [online]. Retrieved from," https://ontotext.com/knowledgehub/fundamentals/information-extraction/, 2018.

[28] V. a. G. S. L. Gupta, "A survey of text summarization extractive techniques.," *Journal of emerging technologies in web intelligence,* vol. 2, no. 0, pp. 258-268, 2010.

[29] Juan M. Madera, "Using social networking websites as a selection tool: The role of selection process fairness and job pursuit intentions.," Conrad N. Hilton College of Hotel and Restaurant Management, University of Houston, 2012.

[30] B. T. J. B. T. G. a. M. Z. Francisco Villarroel Ordenes, "Analyzing Customer Experience Feedback Using Text Mining: A Linguistics-Based Approach.," *Journal of Service Research,* vol. 17, no. 3, 2014.

[31] J. A. a. S. O. R. Roberta Akemi Sinoara, "Text mining and semantics: a systematic mapping study.," *Journal of the Brazilian Computer Society,* 2017.

[32] J. Jarman, "Combining Natural Language Processing and Statistical Text Mining: A Study of Specialized Versus Common Languages.," University of South Florida, 2011.

[33] R. &. W. J. Collobert, "A unified architecture for natural language processing: Deep neural networks with multitask learning.," in *25th international conference on Machine learning*, 2008.

[34] G. S. R. Hinton, " Discovering binary codes for documents by learning deep generative models.," *Topics Cogn Sci,* vol. 3, no. 1, pp. 74-91, 2011.

[35] R. H. E. P. J. M. C. N. A. Socher, "Dynamic pooling and unfolding recursive autoencoders for paraphrase detection.," in *Advances in Neural Information Processing Systems*, 2011.

[36] T. C. K. C. G. &. D. J. Mikolov, "Efficient estimation of word representations in vector space.," in *arXiv preprint arXiv:1301.3781*, 2013.

[37] T. S. I. C. K. C. G. S. &. D. J. Mikolov, "Distributed representations of words and phrases and their compositionality," *In Advances in neural information processing systems,* 2013.

[38] F. a. K. A. Hill, "Learning abstract concept embeddings from multi-modal data: Since you probably can't see what I mean," in *EMNLP*, 2014.

[39] Q. V. &. M. T. Le, "Distributed representations of sentences and documents.," *arXiv preprint arXiv:1405.4053,* 2014.

[40] J. Z. Y. &. Z. Y. Lilleberg, "Support vector machines and Word2vec for text classification with

semantic features.," in *In Cognitive Informatics & Cognitive Computing (ICCI* CC)*, 2015.

[41] Z. L. S. L. Y. L. W. &. J. H. Cao, "A Novel Neural Topic Model and Its Supervised Extension," in *AAAI* , 2016.

[42] K. &. N. H. T. Taghipour, "Semi-Supervised Word Sense Disambiguation Using Word Embeddings in General and Specific Domains.," in *Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 2015.

[43] D. W. F. Q. B. Y. N. L. T. &. Z. M. Tang, " Sentiment Embeddings with Applications to Sentiment Analysis," *IEEE Transactions on Knowledge & Data Engineering,* vol. 28, no. 2, pp. 496-509., 2016.

[44] M. K. X. H. Y. L. L. J. &. F. B. J. Leung, "Deep learning of the tissue-regulated splicing code.," *Bioinformatics,* vol. 30, no. 12.

[45] R. P. K. L. J. D. A. S. A. W. J. Z. Y. Heffernan, "Improving prediction of secondary structure, local backbone angles, and solvent accessible surface area of proteins by iterative deep learning.," Scientific reports, 2015.

[46] S. L. B. &. Y. S. Min, " Deep learning in bioinformatics.," *Briefings in bioinformatics,* 2016.

[47] M. M. V. F. K. T. M. S. N. W. R. &. M. E. Najafabadi, "Deep learning applications and challenges in big data analytics," *Journal of Big Data,* vol. 2, no. 1, 2015.

[48] G. E. &. Z. R. S. Hinton, "Autoencoders, minimum description length and Helmholtz free energy.," in *In Advances in neural information processing systems*, 1994.

[49] Y. M. G. D. Y. &. R. S. Bengio, "Better mixing via deep representations.," in *ICML-13*, 2013.

[50] R. &. H. G. alakhutdinov, "Deep boltzmann machines. In (pp. 448-455).," in *Artificial Intelligence and Statistics* , 2009.

[51] S. S. A. &. B. E. Semeniuta, "A Hybrid Convolutional Variational Autoencoder for Text Generation.," *arXiv preprint arXiv:1702.02390,* 2017.

[52] M. P. M. M. A. a. S. B. Marcus, "Building a large annotated corpus of english: The penn treebank.," *Computational Linguistics ,* vol. 19, no. 2, p. 313–330, 1993.

[53] J. L. M. T. &. J. D. Li, "A hierarchical neural autoencoder for paragraphs and documents.," *arXiv preprint arXiv:1506.01057.,* 2015.

[54] M. S. M. Ranzato, "Semi-supervised learning of compact document representations with deep networks.," in *25th International Conference on Machine Learning. ACM*, 2008.

[55] Z. S. Harris, "Distributional structure," *Word,* vol. 10, no. 2-3, pp. 146-162, 1954.

[56] H. W. N. &. Y. D. Y. Wang, "Collaborative deep learning for recommender systems.," in *21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015.

[57] D. M. N. A. Y. &. J. M. I. Blei, "Latent Dirichlet allocation.," *The Journal of machine learning research,* vol. 3, pp. 993-1022, 2003.

[58] Nisar T.M. and Yeung M.: Twitter as a tool for forecasting stock market movements: A short-window event study, The Journal of Finance and Data Science, https://doi.org/10.1016/j.jfds.2017.11.002. Feb. 2018.

[59] Y. Kim, "Convolutional Neural Networks for Sentence Classification," in *EMNLP*, Doha, Qatar, 2014.

[60] Baziotis, C., Pelekis, N. and Doulkeridis, C. : DataStories at SemEval-2017 Task 4: Deep LSTM with Attention for Message-level and Topic-based Sentiment Analysis. Proc. SemEval-2017, pp. 502-518, Vancouver, Canada, Aug. 2017.

[61] Rosenthal, S., Farra, N. and Nakov, P. : SemEval-2017 Task 4: Sentiment Analysis in Twitter, Proc. SemEval-2017, pp. 502-518, Vancouver, Canada, Aug. 2017.

[62] S. M. &. K. S. Mohammad, "Understanding emotions: A dataset of tweets to study interactions between affect categories," in *11th Edition of the Language Resources and Evaluation Conference*, 2018.

[63] L. W. J. &. C. Y. Deng, " Joint inference and disambiguation of implicit sentiments via implicature constraints," in *COLING*, 2014.

[64] L. &. L. B. Zhang, "Identifying noun product features that imply opinions," in *49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies* , 2011.

[65] I. C. T. &. S. C. Russo, "SemEval-2015 Task 9: CLIPEval Implicit Polarity of Events," in *In Proceedings of the 9th International Workshop on Semantic Evaluation* , 2015.

[66] R. &. M. M.-F. Mochales Palau, "Argumentation mining: the detection, classification and structure of arguments in text," in *12th International Conference on Artificial Intelligence and Law*, 2009.

[67] M. &. T. P. Lippi, "Argumentation Mining: State of the Art and Emerging Trends.," *ACM Transactions on Internet Technology,* vol. 16, no. 2, 2016.

[68] M. &. B. M. Miwa, "End-to-end relation extraction using LSTMs on sequences and tree structures," in *54th Annual Meeting of the Association for Computational Linguistics*, 2016.

[69] H. &. D. M. Schwenk, "Learning joint multilingual sentence representations with neural machine translation.," in *2nd ACL Workshop on Representation Learning for NLP*, 2017.

[70] S. D. J. &. G. I. Eger, "Neural end-to-end learning for computational argumentation mining.," in *55th Annual Meeting of the Association for Computational Linguistics*, 2017.

[71] C. E. S. D. J. K. T. &. G. I. Schulz, "Multi-task learning for argumentation mining in low-resource settings.," in *16th Annual Conference of the North American Chapter of the Association for computational linguistics: Human Language Technologies*, 2018.

[72] H. L. M.-T. &. M. C. D. Pham, "Learning distributed representations for multilingual text sequences," in *NAACL-HLT*, 2015.

[73] H. &. D. M. Schwenk, "Learning joint multilingual sentence representations with neural machine translation," in *2nd ACL Workshop on Representation Learning for NLP*, 2017.

[74] S. T. A. B. Y. &. P. C. J. Subramanian, "Learning general purpose distributed sentence representations via large scale multi-task learning," in *Sixth International Conference on Learning Representations*, 2018.

[75] S. P. G. K. a. B. P. Pawar, " Relation extraction: a survey," *arXiv,* vol. :1712.05191, 2017.

[76] D. A. S. a. E. G. Saeideh Bakhshi, " Faces engage us: Photos with faces attract more likes and comments on instagram," in *Human factors in computing systems*, 2014.

[77] P. G. a. H. Wu., "A cache partition policy of ccn based on content popularity.," in *Advanced Science and Technology*, 2016.

[78] W. Z. M. C. L. S. a. S. Y. Zhi Wang, "Cpcdn: Content delivery powered by context and user intelligence," *IEEE Transactions on Multimedia,* vol. 17, pp. 92-103, 2015.

[79] P. e. a. Zhou, "Attention-based bidirectional long short-term memory networks for relation classification," in *Proc. ACL* , 2016.

[80] T. H. a. G. R. Nguyen, "Relation extraction: Perspective from convolutional neural networks.," in *1st Workshop on Vector Space Modeling for Natural Language Processing.*, 2016.

[81] R. e. a. Socher, "Semantic compositionality through recursive matrix-vector spaces.," in *Proc. EMNLP*, 2012.

[82] M. e. a. Mazloom, "Multimodal popularity prediction of brand-related social media posts,," in *ACM on Multimedia*, 2016.

[83] K. K. S. F. X. J. H. C.-. N. C. a. Y. J. L. Wenjian Hu, "Predicting the image propagation path in online

social networks," 2018.

[84] K. M. B. a. J.-M. F. Wang, "Retweet Wars: Tweet Popularity Prediction via Dynamic Multimodal Regression," in *Applications of Computer Vision (WACV)*, 2018.

[85] X. e. a. Zeng, "Large Scaled Relation Extraction with Reinforcement Learning," in *AAAI*, 2018.

[86] F. 2. T. c. A. Z. (. Sebastiani, " Text Mining and its Applications,," in *WIT Press*, Southampton, UK, 2005.

[87] L. Y. a. P. B. Y. Miao, " Neural Variational Inference for Text Processing.," in *Mcmc*, 2015.

[88] A. Srivastava and C.Sutton., "Autoencoding Variational Inference For Topic Models," 2017.

[89] E. G. a. P. B. Y. Miao, "Discovering Discrete Latent Topics with Neural Variational Inference," 2014.

[90] D. P. K. a. M. Welling., " Auto-Encoding Variational Bayes.," in *ICLR*, 2014.

[91] C. E. Moody., "Mixing Dirichlet Topic Models and Word Embeddings to Make lda2vec.," 2016.

[92] A. A. a. A. J. S. M. Zaheer, "Latent LSTM Allocation: Joint Clustering and Non-Linear Dynamic Modeling of Sequence Data.," in *Icml*, 2017.

[93] C. W. J. G. a. J. P. A. B. Dieng, "TopicRNN: A Recurrent Neural Network with Long-Range Semantic Dependency," 2016.

[94] W. D. S. a. M.-F. Moens., "Cross-language linking of news stories on the Web using interlingual topic modeling," in *CIKM 2009 Workshop on Social Web Search and Mining*, 2009.

[95] H. W. J. N. D. A. S. a. A. M. D. Mimno, "Polylingual topic models," in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2009.

[96] I. V. a. M.-F. M. G. Heyman, " C-BiLDA extracting cross-lingual topics from non-parallel texts by distinguishing shared from unshared content.," *Data Mining and Knowledge Discovery,* vol. 30, no. 5, 2016.

[97] J. J. a. H. D. III., "Extracting Multilingual Topics from Unaligned Comparable Corpora," in *32nd Annual European Conference on Advances in Information Retrieval (ECIR)*, 2010.

[98] M. L. B. B. C. M.-C. M. a. J. M. J. Ioannis Arapakis, "User engagement in online news: Under the scope of sentiment, interest, affect, and gaze., 65(10)," *Journal of the Association for Information Science and Technology,* vol. 65, p. 1988–2005, 2014.

[99] J.-H. O. a. Y. T. Anjana Susarla, " Social networks and the diffusion of user-generated content:

Evidence from youtube.," *Information Systems Research,* vol. 23, pp. 23-41, 2012.

[100] D. A. S. a. E. G. Saeideh Bakhshi, " Faces engage us: Photos with faces attract more likes and comments on instagram.," in *Human factors in computing systems*, 2014.

[101] P. G. a. H. Wu., " A cache partition policy of ccn based on content popularity.," *Advanced Science and Technology,* p. 92:9–16, 2016.

[102] W. Z. M. C. L. S. a. S. Y. Zhi Wang, " Cpcdn: Content delivery powered by context and user intelligence.," *IEEE Transactions on Multimedia,* vol. 17, pp. 92-103, 2015.

[103] K. K. S. F. X. J. H. C.-. N. C. a. Y. J. L. Wenjian Hu, "Predicting the image propagation path in online social networks," 2018.

[104] W.-H. C. Y. Z. Q. H. J. L. a. T. M. Bo Wu, "Sequential prediction of social media popularity with deep temporal context networks.," ArXiv, 2017.

[105] K. M. B. a. J.-M. F. Wang, "Retweet Wars: Tweet Popularity Prediction via Dynamic Multimodal Regression," in *Applications of Computer Vision (WACV)*, 2018.

[106] M. e. a. Mazloom, "Multimodal popularity prediction of brand-related social media posts," in *ACM on Multimedia*, 2016.

[107] M. B. H. a. M. W. Mazloom, "Multimodal Context-Aware Recommender for Post Popularity Prediction in Social Media.," in *ACM Multimedia*, 2017.

[108] G. Z. a. V. S. D. Min Du Feifei Li, " Anomaly Detection and Diagnosis from System Logs through Deep Learning,," in *ACM SIGSAC* , 2017.

[109] L. H. A. F. D. P. a. M. I. J. Wei Xu, "Detecting large-scale system problems by mining console logs," in *ACM Symposium on Operating Systems Principles - SOSP*, 2009.

[110] Q. F. S. Y. J. L. a. B. W. Jian-Guang Lou, "Mining program work.ow from interleaved traces.," in *SIGKDD*, 2010.

[111] P. J. J. X. G. J. H. Z. a. G. J. Xiao Yu, " CloudSeer: Work.ow Monitoring of Cloud Infrastructures via Interleaved Logs," in *ACM International Conference on Architectural Support for Programming Languages and Operating systems*, 2016.

[112] "Log File Anomaly Detection," https://cs224d.stanford.edu/reports/YangAgrawal.pdf.

[113] S. R. S. K. a. C. L. Sarah M. Erfani, " High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning," *Journal Pattern Recognition Archive,* vol. 58, no. C, 2016.

[114] J. K. C. S. K. J. K. Donghwoon Kwon Hyunjoo Kim, "A survey of deep learning-based network anomaly detection,," *Cluster Computing,* 2017.

[115] A. G. R. k. a. R. k. Madhu. G, "Intelligent Semantic Web Search Engines: A Brief Survey," *ijwest,* 2011.

[116] O. a. N. K. Barkan, "Item2vec: neural item embedding for collaborative filtering," in *IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*, 2016.

[117] A. A. D. S. a. R. H. Khosla, "What makes an image popular?," in *WWW*, 2014.

[118] F. e. a. Kooti, "Understanding short-term changes in online activity sessions," in *International World Wide Web Conferences Steering Committee*, 2017.

[119] I. B. B. C. a. M. L. Arapakis, " On the feasibility of predicting popular news at cold start," *Journal of the Association for Information Science and Technology,* 2017.

[120] S. R. F. a. N. C. Mohammadi, "User Reactions Prediction Using Embedding Features," *arXiv,* 2018.

[121] Androniki Sapountzi and Kostas E. Psannis, "Social Networking Data Analysis Tools & Challenges.," in *Future Generation Computer Systems*, 2016.

[122] M. G. M. M.-I. K. M. a. A. W. Stephen Kelley, "Defining and Discovering Communities in Social Networks," 2011.

[123] T. Z. Devavrat Shah, "Community detection in Networks: The Leader-Follower Algorithm.," Department of Electrical Engineering and Computer Science Massachusetts Institute of Technology, 2010.

[124] D. S. a. T. Z. Dhruv Parthasarathy, "Leaders, Followers, and Community Detection,," Massachusetts Institute of Technology, 2017.

[125] R. A. Hanneman and M. Riddle, "Introduction to Social Network Methods," 2005.

[126] Doaa Gamal, " Social Networks Influence Analysis," UNF Theses and Dissertations, 2017.

[127] S. P. J. S. a. H. T. U.Kang, "Centralities in Large Networks: Algorithms and Observations.," Carnegie Mellon University, 2011.

[128] Q. L. E. C. H. X. Y. Z. a. Y. Y. Biao Xiang, "PageRank with Priors: An Influence Propagation Perspective," 2013.

[129] Lise Getoor and Christopher P.Diehl, " Link Mining: A Survey," in *SIGKDD*, 2005.

[130] J. T. L. a. N. V. C. Ryan N. Lichtenwalter, "New Perspectives and Methods in Link Prediction.,"

University of Notre Dame, 2010.

[131] S.Banuchitra and Dr. K. Kunugumaraj, " A Comprehensive Survey of Content Based Image Retrieval Techniques.," *International Journal of Engineering and Computer Science,* vol. 5, no. 8, 2016.

[132] Rebecca Mason and Eugene Charniak, "Domain-Specific Image Captioning.," Brown Laboratory for Linguistic Information Processing (BLLIP), Brown University, 2014.

[133] I. K. a. D. G. Nikolaos Panagiotou, " Detecting Events in Online Social Networks: Definitions, Trends and Challenges.," *Springer International Publishing Switzerland,* 2016.

[134] M. d. G. a. G. S. Pasquale Lops, "Content-based Recommender Systems: State of the Art and Trends.," in *Recommender Systems Handbook*, 2011.

[135] J. T. R. a. J. A. K. Michael D. Ekstrand, " Collaborative Filtering Recommender Systems. Foundations and Trends in Human-Computer Interaction," 2011.

[136] J. M. Y. K. Y. S. Sara Cohen, "XSEarch: a semantic search engine for XML," in *29th international conference on Very large data bases*.

[137] H. K. K. Z. X. M. H. W. Georges Gardarin, "SEWISE: An Ontology-based Web Information Search Engine.," in *NLDB* , 2003.

[138] A. e. a. Hogan, "Searching and browsing linked data with swse: The semantic web search engine.," *Web semantics: science, services and agents on the world wide web,* vol. 9, no. 4, pp. 365-401, 2011.

[139] D. W. M. M. W. W. a. G. Z. J. Lu, "Recommender system application developments: A Survey," *Decis. Support Syst.,* vol. 74, p. 12–32, 2015.

[140] S. L. a. T. Lakshmi, "Recommendation Systems:," *Int. J. Comput. Sci. Inf. Technol.,* vol. 5, p. 5771–5772, 2014.

[141] L. Y. a. A. S. Shuai Zhang, "Deep Learning based Recommender System: A Survey and New Perspectives.," *ACM J. Comput. Cult. Herit.,* vol. 1, p. 35, 2017.

[142] L. L. H. Z. L. N. X. H. a. T.-S. C. Xiangnan He, "Neural collaborative filtering," in *Proceedings of the 26th International Conference on World Wide Web*, 2017.

[143] X. H. L. N. a. T.-S. C. Xiang Wang, ".Item Silk Road: Recommending Items from Information Domains to Social Users," in *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*, 2017.

[144] L. K. J. H. T. S. T. C. H. A. G. A. G. C. W. Heng-Tze Cheng, "Wide & deep learning for recommender systems," in *Proceedings of the 1st Workshop on Deep Learning for Recomender Systems ACM*, 2016.

[145] R. T. Y. Y. Z. L. a. X. H. Huifeng Guo, "DeepFM: A Factorization-Machine based Neural Network for CTR Prediction," *IJCAI,* p. 2782–2788, 2017.

[146] A. K. M. S. S. a. L. X. Suvash Sedhain, "Autorec: Autoencoders meet collaborative filtering," in *Proceedings of the 24th International Conference on World Wide Web. ACM*, 2015.

[147] R. G. a. J. M. Florian Strub, "Hybrid Recommender System based on Autoencoders," in *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems. ACM,*, 2016.

[148] W. L. W. R. a. Z. X. Yuanxin Ouyang, "Autoencoder-based collaborative €ltering," in *International Conference on Neural Information Processing. Springer*, 2014.

[149] C. D. A. X. Z. a. M. E. Yao Wu, "Collaborative denoising auto-encoders for top-n recommender systems," in *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining. ACM*, 2016.

[150] N. W. a. D.-Y. Y. Hao Wang, "Collaborative deep learning for recommender systems," in *n Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015.

[151] L. C. Y. X. a. J. W. Haochao Ying, "Collaborative deep ranking: a hybrid pair-wise recommendation algorithm with implicit feedback," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer*, 2016.

[152] Y. G. a. Q. Zhang, ". Hashtag Recommendation Using AŠention-Based Convolutional Neural Network," *IJCAI,* p. 2782–2788, 2016.

[153] V. N. a. P. S. Y. Lei Zheng, "Joint Deep Modeling of Users and Items Using Reviews for Recommendation," in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining (WSDM '17). ACM*, 2017.

[154] Y. W. J. T. K. S. S. R. a. H. L. Suhang Wang, " What Your Images Reveal: Exploiting Visual Contents for Point-of-Interest Recommendation," in *Proceedings of the 26th International Conference on World Wide Web (WWW '17)*, 2017.

[155] S. D. a. B. S. Aaron Van den Oord, "Deep content-based music recommendation," in *Advances in neural information processing systems*, 2013.

[156] B. Y. Z. Z. J. S. a. H. L. Xiaoxuan Shen, "Automatic Recommendation Technology for Learning Resources with Convolutional Neural Network," in *International Symposium on Educational*

Technology (ISET). IEEE, 2016.

[157] A. K. L. B. a. D. T. Balazs Hidasi, "Session-based recommendations with recurrent neural networks," in *International Conference on Learning Representations*, 2015.

[158] A. A. A. B. A. J. S. a. H. J. Chao-Yuan Wu, "Recurrent recommender networks," in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining. ACM*, 2017.

[159] H. J. a. A. J. Smola., "Neural survival recommender," in *In Proceedings of the Tenth ACM International Conference on Web Search and Data Mining. ACM*, 2017.

[160] T. L. J. J. a. L. Z. Yang Li, "Hashtag recommendation with topical aŠention-based LSTM," in *In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics.*, 2016.

[161] D. B. a. A. M. Trapit Bansal, "Ask the gru: Multi-task learning for deep text recommendations," in *In Proceedings of the 10th ACM Conference on Recommender Systems. ACM*, 2016.

[162] C. C. Œ. L. Y. M. a. X. M. Zhenghua Xu, "Tag-aware personalized recommendation using a deep-semantic similarity model with negative sampling," in *. In Proceedings of the 25th ACM International on Conference on Information and Knowledge Management. ACM*, 2016.

[163] Y. S. a. X. H. Ali Mamdouh Elkahky, "A multi-view deep learning approach for cross domain user modeling in recommendation systems.," in *In Proceedings of the 24th International Conference on World Wide Web. International World Wide Web Conferences Steering Committee*, 2015.

[164] A. M. a. G. H. Ruslan Salakhutdinov, "Restricted Boltzmann machines for collaborative filtering," in *In Proceedings of the 24th international conference on Machine learning. ACM*, 2007.

[165] Y. O. W. R. a. Z. X. Xiaomeng Liu, "Item Category Aware Conditional Restricted Boltzmann Machine Based Recommendation," in *. In International Conference on Neural Information Processing*, 2015.

[166] B. T. W. D. a. H. Z. Yin Zheng, "A Neural Autoregressive Approach to Collaborative Filtering," in *In Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48 (ICML'16). JMLR.org,*, 2016.

[167] L. Y. W. Z. Y. G. Y. X. B. W. P. Z. a. D. Z. Jun Wang, "A Minimax Game for Unifying Generative and Discriminative Information Retrieval Models," in *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*, 2016.

[168] N. J. Y. D. L. X. X. a. W.-Y. M. Fuzheng Zhang, " Collaborative knowledge base embedding for recommender systems," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016.

[169] Y. A. H. L. S. H. a. S.-g. L. Hanbit Lee, "‹ote Recommendation in Dialogue using Deep Neural Network," in *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval ACM*, 2016.

[170] D. L. W. L. Z.-J. Z. a. H. L. Chenyi Lei, "Comparative Deep Learning of Hybrid Representations for Image Recommendations," in *Proceedings of the IEEE Conference on Computer Vision and Paˆern Recognition*, 2016.

[171] S. X. a. D.-Y. Y. Hao Wang, "Collaborative recurrent autoencoder: Recommend while learning to fill in the blanks," in *Advances in Neural Information Processing Systems*, 2016.

[172] Z. W. Z. R. L. B. a. W. L. Piji Li, "Neural Rating Regression with Abstractive Tips Generation for Recommendation," in *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*, 2017.

[173] L. D. M. G. X. H. a. P. P. Jianfeng Gao, "Modeling interestingness with deep neural networks," US Patent App. 14/304,863., June 13 2014.

[174] A. M. E. a. X. H. Yang Song, "Multi-rate deep learning for temporal recommendation," in *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval. ACM*, 2016.

[175] Haipeng Shen, "Statistical Analysis of Call-Center Operational Data: Forecasting Call Arrivals, and Analyzing Customer Patience and Agent Service," https://doi.org/10.1002/9780470400531.eorms0827.

[176] "The 2017 Voice Report by VoiceLabs," http://voicelabs.co/2017/01/15/the-2017-voice-report/, 2017.

[177] "Voice Payments Emerge as Tech Giants Compete for Voice-First Commerce," https://thefinancialbrand.com/66046/voice-payments-banking-ai-amazon-alexa-siri, 2017.

[178] Genesys, " "Omnichannel customer experience – cloud & on premise,"," http://www.genesys.com/. Accesat: Mar. 6, 2017, 2017.

[179] Avaya Inc, ""Customer & team engagement solutions – business communication – Avaya,"," http://www.avaya.com/en/. Accesat: Mar. 6, 2017, 2017.

[180] A. J. A. N. a. Alex Fandrianto, "Speaker Recognition Using Deep Belief Network," CS, 229, 2012.

[181] Y. L. a. L. F. M. McLaren, ""Advances in deep neural network approaches to speaker recognition,"," in *40th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015.

[182] L. M. Rojas-Barahona, "Deep learning for sentiment analysis".

[183] BoWang MinLiu, "Deep Learning for Aspect-Based Sentiment Analysis".

[184] J. Levashina, " Expected practices in background checking: review of the human resource management literature.," *Employee Responsibilities and Rights Journal,* vol. 21, p. 231–241, 2009.

[185] Ricardo Buettner, " A Framework for Recommender Systems in Social Network Recruiting: An Interdisciplinary Call to Arms.," in *47th Hawaii International Conference on System Science*, 2014.

[186] S. Y. M. Al-Otaibi, " A survey of job recommender systems," *International Journal of the Physical Sciences ,* vol. 7, no. 29, p. 5127–5142, 2012.

[187] J. a. K. T. a. W. O. a. W. T. Malinowski, "Matching people and jobs: A bilateral recommendation approach," in *System Sciences, 2006. HICSS'06. Proceedings of the 39th Annual Hawaii International Conference on*, 2006.

[188] A. a. K. E. Golec, "A fuzzy model for competency-based employee evaluation and selection," *Computers \& Industrial Engineering,* vol. 52, no. 1, pp. 143--161, 2007.

[189] I. a. C. B. B. a. G. A. Paparrizos, Machine learned job recommendation, ACM, 2011, pp. 325--328.

[190] L. a. F. W. a. W. L. Zhang, "Pj matching model of knowledge workers," *Procedia Computer Science,* vol. 60, pp. 1128--1137, 2015.

[191] R. a. S. J. Feldman, The text mining handbook: advanced approaches in analyzing unstructured data, Cambridge university press, 2007.

[192] G. a. H. M. a. o. Berio, "Knowledge management for competence management," *Journal of Universal Knowledge Management,* vol. 1, pp. 21--28, 2005.

[193] J. a. N. T. Dorn, "Meta-search in human resource management," *International Journal of Human and Social Sciences,* vol. 1, no. 2, 2006.

[194] W. Ben Abdessalem Karaa, "Web-based recruiting: A framework for cvs handling," in *Second international conference on web and information technologies" ICWIT*, 2009, pp. 12--14.

[195] A. J. a. N. S. a. O. J. Gill, What Are They Blogging About? Personality, Topic and Motivation in Blogs., ICWSM, 2009.

[196] G. a. o. Mishne, "Experiments with mood classification in blog posts," in *Proceedings of ACM SIGIR 2005 workshop on stylistic analysis of text for information access*, vol. 19, 2005, pp. 321--327.

[197] J. W. a. K. L. A. Pennebaker, "Linguistic styles: Language use as an individual difference.," *Journal of personality and social psychology,* vol. 77, no. 6, p. 1296, 1999.

[198] P. Ekman and H. Oster, "Facial expressions of emotion," *Annu. Rev. Psychol.,* no. 30, p. 527–554, 1979.

[199] C. Shan, S. Gong and P. W. McOwan, "Facial expression recognition based on local binary patterns: A comprehensive study," *IVC,* vol. 27, no. 6, p. 803–816, 2009.

[200] G. Littlewort, M. S. Bartlett, I. Fasel, J. Susskind and J. Movellan, "Dynamics of facial expression extracted automatically from video," *CVPR Workshops,* 2004.

[201] A. Dhall, A. Asthana and R. G. a. T. Gedeon, "Emotion recognition using PHOG and LPQ features," *FG,* p. 878–883, 2011.

[202] O. R. V. P. a. M. P. R. Walecki, "Variable-state latent conditional random fields for facial expression recognition and action unit detection," *FG,* p. 1–8, 2015.

[203] H. Huang and T. Tang, "3D facial expression recognition based on automatically selected features," *CVPR,* p. 1–8, 2008.

[204] S. Berretti, B. B. Amor and M. D. a. A. D. Bimbo, "3D facial expression recognition using sift descriptors of automatically detected keypoints," *TVC,* vol. 27, no. 11, p. 1021–1036, 2011.

[205] A. Dapogny, K. Bailly and a. S. Dubuisson, "Dynamic facial expression recognition by joint static and multi-time gap transition classification," *FG,* 2015.

[206] B. Sun, L. Li, T. Zuo, Y. Chen and G. Z. a. X. Wu, "Combining multimodal features with hierarchical classifier fusion for emotion recognition in the wild," *ICMI,* p. 481–486, 2014.

[207] S. Koelstra and M. P. a. I. Patras, "A Dynamic Texture-Based Approach to Recognition of Facial Actions and Their Temporal Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 32, no. 11, pp. 1940-1954, 2010.

[208] B. Allaert, I. M. Bilasco and C. Djeraba, "Consistent Optical Flow Maps for Full and Micro Facial Expression Recognition.," *VISAPP,* pp. 235-242, 2017.

[209] I. Song, H. J. Kim and P. B. Jeon, "Deep learning for real-time robust facial expression recognition on a smartphone," *ICCE,* p. 564–567, 2014.

[210] M. Liu, S. Li and S. S. a. X. Chen, "Au-aware deep networks for facial expression recognition," *FG,* p. 1–6, 2013.

[211] S. He, S. Wang, W. Lan, H. Fu and Q. Ji, "Facial expression recognition using deep boltzmann

machine from thermal infrared images," *ACII,* p. 239–244, 2013.

[212] G. K. Verma and U. S. Tiwary, "Multimodal fusion framework: A multiresolution approach for emotion classification and recognition from physiological signals," *NeuroImage,* vol. 102, no. 1, pp. 162-172, 2014.

[213] S. Poria, H. Peng, A. Hussain, N. Howard and E. Cambria, "Ensemble application of convolutional neural networks and multiple kernel learning for multimodal sentiment analysis," *Neurocomputing,* vol. 261, pp. 217-230, 2017.

[214] S. Poria, E. Cambria, R. Bajpai and A. Hussain, "A review of affective computing: From unimodal analysis to multimodal fusion," *Information Fusion,* vol. 37, pp. 98-125, 2017.

[215] Z. Yin, M. Zhao, Y. Wang, J. Yang and J. Zhang, "Recognition of emotions using multimodal physiological signals and an ensemble deep learning model," *Computer Methods and Programs in Biomedicine,* vol. 140, pp. 93-110, 2016.

[216] A. Eitel, J. T. Springenberg, L. Spinello, M. A. Riedmiller and W. Burgard, "Multimodal deep learning for robust RGB-D object recognition," *IROS 2015,* pp. 681-687.

[217] V. L. G. S. e. a. Gupta, " A survey of text mining techniques and applications," *Journal of emerging technologies in web intelligence ,* vol. 1, no. 1, p. 60–76, 2009.

[218] Xu, W., Huang, L., Fox, A., Patterson, D., & Jordan, M. I., "Detecting large-scale system problems by mining console logs," in *ACM SIGOPS 22Nd Symposium on Operating Systems Principles*, NY, USA, 2009.

[219] Liang, Y., Zhang, Y., Xiong, H., & Sahoo, R., "Failure Prediction in IBM BlueGene/L Event Logs.," *IEEE. ICDM,* 2007

[220] X. e. al., ""Detecting large-scale system problems by mining console logs",," *in ACM,* 2009.

[221] Bagnall, "The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances," *Data Mining and Knowledge Discovery ,* vol. 31, no. 3, pp. 606-660, 2017.

[222] Makridakis, "The accuracy of machine learning (ml) forecasting methods versus statistical ones: Extending the results of the m3-competition," 2017.

[223] S. F. H. M. a. N. K. Crone, "Advances in forecasting with neural networks? empirical evidence from the nn3 com petition on time series prediction.," *International Journal of forecasting,* vol. 27, no. 3, p. 635–660, 2011.

[224] Nguyen, "A survey of techniques for internet traffic classification using machine learning," *IEEE*

Communications Surveys & Tutorials , vol. 10, no. 4, pp. 56-76, 2008.

[225] Lee, "Proactive thermal-aware virtual machine allocation in hpc cloud datacenters.," in *International Conference on High Performance Computing (HiPC)*, 2012.

[226] Pinheiro, "Failure trends in a large disk drive population.," *FAST* , vol. 7, pp. 17-23, 2007.

[227] Li, "Hard drive failure prediction using classification and regression trees.," in *Dependable Systems and Networks (DSN), 44th Annual IEEE/IFIP International Conference*, 2014.

[228] Hsu, C. H., & Feng, W. C., "Reducing overheating-induced failures via performance-aware CPU power management," in *6th International Conference on Linux Clusters: The HPC Revolution*, 2005.

[229] InformationWeek, "Big Data Analytics: Descriptive Vs. Predictive Vs. Prescriptive," https://www.informationweek.com/big-data/big-data-analytics/big-data-analytics-descriptive-vs-predictive-vs-prescriptive/d/d-id/1113279, 2018.

[230] B. Liu, "Sentiment analysis and opinion mining: Synthesis Lectures on Human Language Technologies," *Morgan & Claypool Publishers,* 2016.

[231] S. M. a. J. B. T. Vohra, "A comparative study of sentiment analysis techniques," *Journal JIKRCE,* vol. 2, no. 2, pp. 313-317, 2013.

[232] G. R.-S. S. O. R. Boyan Bonev, "Opinum: statistical sentiment analysis for opinion classification.," in *Proceedings of the 3rd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis*, 2012.

[233] Srivastava, A., & Sutton, C., " Autoencoding Variational Inference For Topic Models," 2017.