TextMine '23

Atelier sur la Fouille de Textes

Organisateurs :

Pascal Cuxac (INIST - CNRS),
Cédric Lopez (Emvista),
Kévin Cousot (Emvista),
Vincent Lemaire (Orange Labs)

Organisé conjointement à la conférence EGC
(Extraction et Gestion des Connaissances)
le 17 janvier 2023 à Lyon

Editeurs :

Pascal Cuxac - INIST - CNRS
2 rue Jean Zay, CS 10310, 54519 Vandoeuvre les Nancy Cedex
Email : pascal.cuxac@inist.fr

Cédric Lopez - Emvista
Cap Oméga, Rond-Point Benjamin Franklin, CS 39521, 34960 Montpellier Cedex 02
Email : cedric.lopez@emvista.com

Vincent Lemaire - Orange Labs
2 avenue Pierre Marzin, 2300 Lannion
Email : vincent.lemaire@orange.com

Publisher:

Vincent Lemaire, Pascal Cuxac, Cédric Lopez
2 avenue Pierre Marzin
22300 Lannion

Lannion, France, 2022

PRÉFACE

C'est une évidence que de dire que nous sommes entrés dans une ère où la donnée textuelle sous toute ses formes submerge chacun de nous que ce soit dans son environnement personnel ou professionnel : l'augmentation croissante de documents nécessaires aux entreprises ou aux administrations, la profusion de données textuelles disponibles via Internet, le développement des données en libre accès (OpenData), les bibliothèques et archives en lignes, les medias sociaux ne sont que quelques exemples illustrant l'évolution de la notion de texte, sa diversité et sa prolifération.

Face à cela les méthodes automatiques de fouille de données (data mining), et plus spécifiquement celles de fouille de textes (text mining) sont devenues incontournables. Récemment, les méthodes de deep learning ont créées de nouvelles possibilités de recherche pour traiter des données massives et de grandes dimensions. Cependant, de nombreuses questions restent en suspens, par exemple en ce qui concerne la gestion de gros corpus textuels multi-thématiques. Pouvoir disposer d'outils d'analyse textuelle efficaces, capables de s'adapter à de gros volumes de données, souvent de nature hétérogène, rarement structurés, dans des langues variées, des domaines très spécialisés ou au contraire de l'ordre du langage naturel reste un challenge.

La fouille de textes couvre de multiples domaines comme, le traitement automatique des langues, l'intelligence artificielle, la linguistique, les statistiques, l'informatique et les applications sont très diversifiées, que ce soit la recherche d'information, le filtrage de spam, le marketing, la veille scientifique ou économique, la lutte antiterroriste...

Le but de cet atelier est de réunir des chercheurs sur la thématique large de la fouille de textes. Cet atelier vise à offrir une occasion de rencontres pour les universitaires et les industriels, appartenant aux différentes communautés de l'intelligence artificielle, l'apprentissage automatique, le traitement automatique des langues, pour discuter des méthodes de fouille de textes au sens large et de leurs applications.

P. Cuxac — INIST-CNRS
C. Lopez — Emvista
V. Lemaire — Orange Labs

Membres du comité de lecture

Le Comité de Lecture est constitué de:

Vincent Claveau (IRISA, Rennes)

Kevin Cousot (Emvista, Montpellier)

Nicolas Dugué (LIUM, Le Mans)

Natalia Grabar (STL - Lille3, Lille)

Adrien Guille (ERIC, Univ. Lyon2, Lyon)

Jean-Charles Lamirel (LORIA, Nancy)

Denis Maurel (Lifat, Université F. Rabelais, Tours)

Ellouze Mourad (MIRACL Laboratory, FSEGS, Sfax, Tunisie)

Anna Pappa (LIASD, Univ. Paris 8, Paris)

Thibault Prouteau (LIUM, Le Mans)

Solen Quiniou (LS2N, Nantes Université, Nantes)

David Reymond (Université de Toulon, Toulon)

Andon Tchechmedjiev (IMT, Alès)

TABLE DES MATIÈRES

Exposé Invité

Graph Neural Networks for NLP
Adrien Guille 1

Session Exposés

Improving Convolutional Neural Networks for Knowledge Graph Completion
Kevin Cousot, Nilofar Moradi Farisar, Waleed Ragheb, Mehdi Mirzapour 3

Une contribution du Text-mining à la connaissance du langage des cachalots
José Coch, Lara Berkenbaum, Olivier Adam 15

Traitement automatique de la coréfénce dans les textes économiques : un exemple d'application chez ReportLinker
Marilyne Latour, Paul Moncuquet 33

LoGE: Expansion Locale-Globale de document non supervisée avec un moteur de recherche Extensible
Oussama Ayoub, Ludovic Li, Christophe Rodrigues, Nicolas Travers 41

Session Défis

Défi TextMine'23 - Reconnaissance d'entités d'intérêts dans les signatures d'e-mails
Kévin Cousot, Cédric Lopez, Pascal Cuxac, Vincent Lemaire 45

Participation d'EDF RD au Défi Textmine 2023 : Reconnaissance d'entités d'intérêts dans les signatures
Philippe Suignard, Leila Hassani, Meryl Bothua 51

GREYC@TextMine2023 : Reconnaissance d'entités nommées dans les signatures d'e-mails
Tanguy Gernot, Emmanuel Giguet 61

Hybridation des approches symboliques et apprentissage profond pour la reconnaissance des entités dans les signatures de mail
Duc Hau Nguyen, Nicolas Fouqué, Victor Klötzer, Hugo Thomas 71

Nextino@TextMine'23 : Approche hybride pour la reconnaissance d'entités d'intérêt dans les signatures d'e-mail
Maëlle Brassier, Asceline Goudjo 75

Index des auteurs **87**

Graph Neural Networks for NLP

Adrien Guille
Laboratoire ERIC, Université Lumière Lyon 2, Lyon

adrien.guille@univ-lyon2.fr

Résumé :

The last couple of years have seen graph neural networks (GNNs) emerging as powerful tools for NLP; enlarging an already wide set of techniques available to practitioners, such as convolutional or recurrent neural networks and pre-trained language models based on the Transformer architecture. In this talk, I'll discuss how to formulate classical NLP problems in terms of graphs and how to solve them efficiently with GNNs. I'll also discuss some research paths opened by this novel approach, in connection with my own work: sparse structure learning, hyperbolic embeddings, etc.

Improving Convolutional Neural Networks for Knowledge Graph Completion

Kevin Cousot*, Nilofar Moradi Farisar**
Waleed Ragheb***,**** Mehdi Mirzapour****

*Emvista, 10 Rue Louis Breguet Le 610, 34830 Jacou
kevin.cousot@emvista.com,
https://www.emvista.com/
**Tarbiat Modares University, Tehran Jalal AleAhmad Nasr
nilofar.moradi.farisar@gmail.com
https://www.modares.ac.ir
***FCAI Cairo University, 5 Dr. Ahmed Zewail Street, Orman, Giza, Egypt
waleed.ragheb@lirmm.fr
https://fci.cu.edu.eg/Home
****LIRMM, 161 Rue Ada, 34095 Montpellier
mehdi.mirzapour@lirmm.fr,
https://www.lirmm.fr/

Abstract. In this paper, we present our ongoing research on a convolutional neural network architecture for knowledge graph completion task called Multi-layered ConvKB (MConvKB). Our new proposal is an extension to ConvKB with two main changes: (i) we added non-linear fully connected layers motivated by VGG architectures which acts as the last layer classifier, and (ii) we initialized the model with different pre-trained translational knowledge graph embeddings (TransE and TransD) to explore its effect in the model performance. Initialized model with TransD has shown better results. Experiments show that MConvKB can achieve significant results just by adding these two simple and yet effective modifications to the previous ConvKB model. To evaluated our approach we used RezoJDM16k, a relation prediction dataset for French. Our experiment show around 10% improvement compared to the original ConvKB model.

1 Introduction

Knowledge Graphs (KGs) are structures of semantic information often represented as multi-relational graphs with nodes and different types of edges. In a KG, each relation is a triple in the form (head (h), relation (r), tail (t)). For instance, the triple (*hunt* $\xrightarrow{r\_agent}$ *tiger*) indicates that *tiger* is an agent of the action *hunt*. There are famous KGs such as Freebase (Bollacker et al., 2008), DBpedia (Lehmann et al., 2015) and WordNet (Miller, 1995) which contain large number of entities and relations. KGs, have been successfully used as valuable resources in many applications such as information retrieval and question answering. Some

studies have also shown (Mahesh and Nirenburg, 1997) natural language processing (NLP) tasks– such as document classification, and machine translation– demand underlying meaning of a text and not just the surface forms. In order to do so, an NLP system must have, in principle, a significant amount of knowledge about the external world represented in Knowledge Graphs. We believe our research is aligned with studies (Mahesh and Nirenburg, 1997) that suggest Knowledge-Based NLP System (KB-NLP) relying on explicitly formulated domain or world knowledge. They can solve typical problems in NLP such as ambiguity resolution and inferencing.

Real-world data are often dynamic and evolving, which makes it hard to build complete resources (Cai et al., 2018; Arora, 2020). There is always missing information (links) between entities, a problem known as KG incompleteness (Wang et al., 2021). *Relation Prediction* or *Knowledge Graph Completion* aims at predicting an absent relation or entity in an incomplete triple. One of the successful approaches to address the relation prediction problem is based on *Knowledge Graph Embedding* (KGE) methods which transform KGs into a low-dimensional vector space while preserving the structure of the KGs and their underlying semantics (Wang et al., 2021).

KGE models are categorized into three different types: translation, semantic and neural network based. Models such as TransE (Bordes et al., 2013a), TransH (Wang et al., 2014), TransR (Lin et al., 2015) and TransD (Ji et al., 2015) are common examples of translational-distance (or additive) models. In TransE, KG relations are considered translational vectors. A triple (h, r, t) is translated from the head entity h to the tail entity t by the relation r. The scoring function $\psi(e_o, r, e_s)$, defined as $-||h+r-t||_2^2$, measures how incorrect a triple is in the embedding space (e_o, r, e_s). In other words, $h+r$ should be close to t. TransH extends TransE by giving each relation its own hyperplane and projects the entities onto it, thus allowing representation of one-to-many, many-to-one and many-to-many relationships. TransD creates a dynamic matrix for all entity-relation pairs and maps the head and tail into M1 and M2, respectively. The transition from head to tail is as follow:

$$M_r^1 = w_r w_h^\perp + I \ , \ M_r^2 = w_r w_t^\perp + I$$
$$h_\perp = M_r^1 h \ , \ t_\perp = M_r^2 t$$
$$\psi(e_o, r, e_s) = -||h_\perp + r - t_\perp||_2^2$$

While two entities can be close with respect to a specific relation, they might simultaneously be distant with respect to another. To address this, TransR uses multiple relation spaces, hence learning relation-specific vector to model entities. Models such as DistMult (Yang et al., 2014), RESCAL (Nickel et al., 2011), ComplEx (Trouillon et al., 2016), and DensE (Lu and Hu, 2020) are well-known Semantic-matching-based (or multiplicative) KGEs. Despite their advantages, these models fail to provide deeper semantics because they ignore hierarchical relationships. Authenticity is measured by using a scoring function that embeds entities and relations into a unified continuous vector space.

Contrary to previous techniques, neural-network-based models or neural tensor network (NTN) (Socher et al., 2013) such as ConvE (Dettmers et al., 2018), ConvKB (Nguyen et al.,

2017), HypER (Balažević et al., 2019), and SACN (Shang et al., 2019) implement deep learning structures that assess temporal features, path information, and structural information in order to produce better embeddings for downstream tasks. We also have specialized graph-based structures such as Graph Neural Nets (GNNs) (Schlichtkrull et al., 2018) and Graph Convolution Networks (GCNs) (Zhang et al., 2019) which have been adopted to improve the expressive power of entity embeddings. GNNs capture the topological features of the entities such as shapes of the neighborhood sub-graphs which are overlooked by the traditional KG embedding methods.

Model	Score function
TransE	$\|\|v_h + v_r - v_t\|\|_2^2$
TransD	$\|\|h_\perp + v_r - t_\perp\|\|_2^2$
ConvKB	$concat(g([v_h, v_r, v_t] * \Omega)) \cdot w$
MConvKB	$concat(g_1(ReLU(g_2(ReLU(g_3([v_h, v_r, v_t] * \Omega)))))) \cdot w$

TAB. 1 – *Models score functions: with g_i as non-linear functions, Ω as convolutional filters, w as weights; $(*)$ and $(\cdot)$ are convolutional and dot operators, respectively.*

In this paper, we present our ongoing research on a convolutional neural network architecture for knowledge graph completion task called Multi-layered ConvKB (MConvKB). Our new proposal is an extension to ConvKB with two main changes: (i) we added non-linear fully connected layers motivated by VGG architectures which acts as the last layer classifier, and (ii) we initialized the model with different pre-trained translational knowledge graph embeddings (TransE and TransD) to explore its effect in the model performance. Initialized model with TransD has shown better results. Experiments show that MConvKB can achieve significant results just by adding these two simple and yet effective modifications to the previous ConvKB model. To evaluated our approach we used RezoJDM16k, a relation prediction dataset for French. Our experiment show around 10% improvement compared to the original ConvKB model. In order to be more precise, we explain the general ideas in our architecture as follow: each element of a triple is associated with its ℓ-dimensional vector representation. Following ConvKB (Nguyen et al., 2017), these vectors are concatenated to obtain the entire triple's embeddings as a $\ell \times 3$ matrix and a convolutional block applies filters to the triple. Extracted features are then concatenated into a single feature vector and processed into a VGG-inspired linear block (Nguyen et al., 2017). Finally, the model outputs a score determining the triple's existence.

We introduce MConvKB a new graph embedding model for knowledge base completion extended from ConvKB. While preserving the properties of ConvKB which generalizes transitional characteristics in transition-based embedding models. We have also increased the performance of the model by adding VGG-inspired linear blocks and trying different initialized graph embeddings from pre-trained translational models such as TransE and TransD. We then evaluate MConvKB on the French RezoJDM16k (Mirzapour et al., 2022) benchmark dataset. Experimental results show that MConvKB obtains better link prediction performance, gaining

a 10% improvement compared to ConvKB.

The rest of the paper is organized as follows: Section 2 describes our system architecture. In Section 3, we describe the system setup for evaluation, followed by a discussion of the results. We conclude our paper in Section 4. It is worth mentioning that the code[1] and RezoJDM16k[2] dataset are publicly available.

2 Architecture

As discussed in section 1, a knowledge graph is composed of a list of triples each in the form (h, r, t). Embedding-based graph models formulate the link prediction task as defining a scoring function that predicts lower scores for the highly probable relations and vice versa. The proposed model is inspired by the ConvKB (Nguyen et al., 2017) architecture that applies a single convolution and linear step for the same task. We introduced the Multilayers ConvKB (MConvKB) model that extends ConvKB by add a VGG-inspired linear classifier module. In our model, we consider the embedding of the triples (v_h, v_r, v_t) each with size ℓ (embedding size). As shown in Figure 1, the model starts by concatenating the triple's embedding to form the matrix $\mathbf{v} = [v_h, v_r, v_t]$. Therefore, $\mathbf{v}$ could be viewed as small images with size ($\ell \times 3$). The convolutional block intakes the matrix, acting as the feature extraction part of the model.

Like ConvKB, we have used only one convolution module. Our experiments with three wide convolution modules (1×3, 3×3 , 5×3), to process the triple's embedding matrix, showed no significant improvement. Having wider VGG-inspired classifier layers will permit the model to inspect the global features. Furthermore, this will help the model to generalize better from the transitional characteristics of back-end embedding models. Following ConvKB, we use batch normalization (Ioffe and Szegedy, 2015), dropout (Srivastava et al., 2014) and ReLU as the nonlinear activation function in the convolutional module.

The feature maps are then processed into a linear module composed of three linear fully connected layers which are inspired by the last non-linear fully connected layers in VGG architectures (Simonyan and Zisserman, 2014). Likewise the convolution block, the linear layers involve dropout and ReLU for the activation function. The final output of the model is the predicted score (s) that indicates the presence of the corresponding relation between the head and the target entities such that:

$$s = \psi_\theta(h, r, t) \tag{1}$$

Where θ corresponds to the train model parameters for both convolutional and linear blocks. The target scores ($\hat{s}$) are induced from the original graph ($\mathcal{V}$) and its corrupted counterpart ($\mathcal{V}'$). This version contains only some random invalid triples.

$$\hat{s}(h, r, t) = \begin{cases} 1 & (h, r, t) \in \mathcal{V} \\ -1 & (h, r, t) \in \mathcal{V}' \end{cases} \tag{2}$$

[1] https://github.com/Emvista/MConvKB

[2] https://github.com/ContentSide/French_Knowledge_Graph

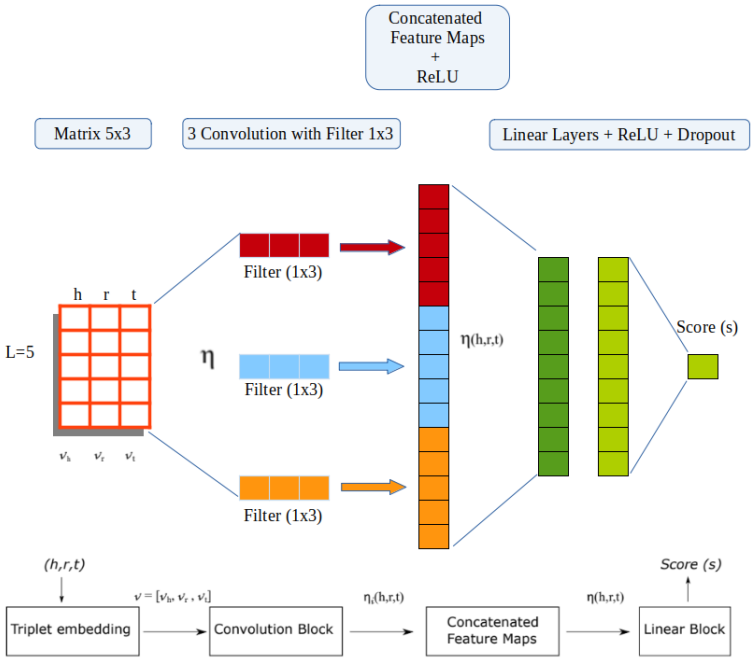


FIG. 1 – *Proposed model architecture*

The model is trained to minimize the loss function ($\mathcal{L}$) in equation 3 over all the training triples in the original and corrupted graphs.

$$\mathcal{L} = \sum_{(h,r,t)\in\{\mathcal{V}\cup\mathcal{V}'\}} \log(1+\exp(s*\hat{s})) \tag{3}$$

Score functions of our model compared to ConvKB and used translational models are listed table 1.

3 Experiments

This section describes the conducted experiments with MConvKB. Firstly, we introduce RezoJDM16k, the benchmark dataset used for evaluation. We briefly mention the initializing graph embeddings before explaining the training protocol. Finally, we discuss the obtained results.

3.1 Dataset

In our experiments, we have used RezoJDM16k (Mirzapour et al., 2022) as a benchmark dataset. RezoJDM (Lafourcade, 2007) is a wide-coverage lexico-semantic network for French

focused on commonsense built through Games With A Purpose (GWAPs), direct contributions, and several inference mechanisms. A relation prediction dataset, RezoJDM15k was extracted from this network (Cousot et al., 2019) but it was poorly suited for knowledge graph embeddings as it was originally designed to be featured engineered from Node2Vec. Furthermore it was not correctly splitted and contained symmetric relations.

Resource	**Entities**	**triples**	**Types**
RezoJDM16k (Train)	$16k$	$666k$	53
RezoJDM16k (Validation)	$16k$	$83k$	53
RezoJDM16k (Test)	$16k$	$83k$	53

TAB. 2 – *Dataset statistics of the RezoJDM16k split*

This led to a subsequent research work that used another methodology to produce RezoJDM16k with the use of a set of filtering criteria and a subgraph selection algorithm (Mirzapour et al., 2022). As illustrated in Table 2, RezoJDM16k contains 16k entities and 53 relation types. Its training set has 666k examples, whereas both validation and testing contain 83k.

3.2 Experimental setup

ConvKB is capable to learn graph embeddings in two ways: (i) learning by randomly initializing the parameters, and (ii) initializing it with already pre-trained models. While ConvKB has only used TransE as the initialized embeddings we have tried to test our models by initializing two translational embedding models, namely TransE and TransD, to see the effect of initialized embedding in the training process.

Both architectures, ConvKB and MConvKB were trained in the same way. AdaGrad (Duchi et al., 2011) was used as an optimizer, with a learning rate of 0.01 and we set L2 regularization parameters to $\lambda = 0.2$ and $\lambda_2 = 0.01$. The training was done on 50 epochs with 500 batches and a negative entity sampling rate set to 10. The hidden layer size matches the embeddings dimension, i.e. 200. The dropout was set to 0.5 and 64 filters were used in the convolutional layer.

3.3 Evaluation

In this section, we compare our proposal, MConvKB, with other models using TransE and TransD initialized embeddings. We follow the existing literature to use major metrics for measuring the quality of embedding models, namely, Hits@K, MR, and MRR (Chen et al., 2020). The following metrics are frequently used and are simple:

Hits@K: is a performance index that measures the probability to find the correct prediction in the first top K model predictions (Chen et al., 2020). By convention, K values vary between 1, 3, 5, and 10. The larger *Hits@K* values are the better predictive performances.

Mean Rank (MR): is the average ranking position of the items predicted by the model among all the possible items (Chen et al., 2020). The smaller the value, the better the model.

Mean Reciprocal Rank (MRR): is the average of the rank inverse for correct predictions. The larger the index, the better the model.

To conduct our evaluation, we have used three scales of Hits@K, i.e. Hits@10, Hits@3, and Hits@1 with other MRR and MR metrics. As reported in Bordes et al. (2013b) and used in the original paper, metric computation is done under the filtered setting to avoid test cases where a corrupted triple is correct and present in the dataset.

Model	**MMR**	**MR**	**H@10**	**H@3**	**H@1**
TransD	0.208	189.186	0.474	0.278	0.064
ConvKB	0.218	186.653	0.493	0.275	0.078
MConvKB	**0.337**	**158.275**	**0.590**	**0.384**	**0.218**

TAB. 3 – *Evaluation results on RezoJDM16k with TransD embeddings*

Model	**MMR**	**MR**	**H@10**	**H@3**	**H@1**
TransE	0.179	203.310	0.432	0.242	0.041
ConvKB	0.202	224.951	0.444	0.247	0.081
MConvKB	**0.295**	**174.763**	**0.536**	**0.330**	**0.183**

TAB. 4 – *Evaluation results on RezoJDM16k with TransE embeddings*

Table 3 and 4 show our evaluation results on RezoJDM16k dataset with TransD and TransE embeddings, respectively. The first row of the mentioned tables shows the initialized embeddings, TransD and TransE, with 0.47 and 0.43 on *Hits@10* scores. The ConvKB model shows around 1.9% and 1.2% improvement of *Hits@10* score over initial TransD and TransE embeddings, respectively. This indicates a slight improvement that is performed by ConvKB. Our proposal MConvKB indicates around 9.7% and 9.2% improvement of *Hits@10* score over TransD and TransE embeddings. This improvement compared to ConvKB is significant and demonstrates the efficiency of trying different initialized embeddings and deep non-linear fully connected layers. While the status of *Hits@3* and *MMR* scores are almost the same comparing to *Hits@10* score in our proposal, *Hits@1* shows significant improvement around 20% for TransD embeddings. We can also observe slight improvement for *MR* in our model.

Table 5 shows the comparison of the performances of KGE state-of-the-art models trained on RezoJDM16k using the evaluation metrics *MRR*, *MR*, *Hits@10*, *Hits@3* and *Hits@1*. In general, we observe that the performance scores of KGE models range from 0.432 to 0.590 for *Hits@10* score. MConvKB has the best performance considering the mentioned evaluation metrics. The second-best performance is for ComplEx with 0.533 *Hits@10* score, just around 6% below MConvKB model performance. This is expected due to the complexity of semantic-based models. TransH and ConvKB models have rather the same result on *Hits@10* metric. The most important observation is the superiority of MConvKB model over all semantic-based

KGE models (DisMult and ComplEx), the translational-based KGE models (TransE, TransH & TransD), and ConvKB as the CNN-based model.

Model	**MRR**	**MR**	**H@10**	**H@3**	**H@1**
TransE*	0.179	203.31	0.432	0.242	0.041
TransH*	0.218	177.12	0.498	0.291	0.069
TransD	0.208	189.19	0.474	0.278	0.064
DistMult*	0.220	194.47	0.445	0.252	0.109
ComplEx*	0.253	201.58	0.533	0.304	0.117
ConvKB	0.218	186.65	0.493	0.275	0.078
MConvKB	**0.337**	**158.27**	**0.590**	**0.384**	**0.218**

TAB. 5 – *Overall performance of knowledge graph embedding models for RezoJDM16k. The results in (*) are from (Mirzapour et al., 2022).*

4 Conclusion and Future Work

In this paper, we presented our ongoing research on a convolutional neural network architecture for knowledge graph completion task called Multi-layered ConvKB (MConvKB). Our new proposal is an extension to ConvKB with two main changes: (i) we added non-linear fully connected layers motivated by VGG architectures which acts as the last layer classifier, and (ii) we initialized the model with different pre-trained translational knowledge graph embeddings (TransE and TransD) to explore its effect in the model performance. Initialized model with TransD has shown better results. Experiments show that MConvKB can achieve significant results just by adding these two simple and yet effective modifications to the previous ConvKB model. To evaluated our approach we used RezoJDM16k, a relation prediction dataset for French. Our experiment show around 10% improvement compared to the original ConvKB model.

For future work and in order to test the generalization of our proposal, we will evaluate our model with two standard English knowledge graph datasets, namely WN18RR (Dettmers et al., 2018) and FB15K-237 (Toutanova and Chen, 2015). Another possible extension to improve current results is testing two characteristics of the network: depth by stacking up several convolution blocks, and possibly wideness by adding more filters of different dimensions. From a practical point of view, having a rich semantic link prediction model for the French language can pave the way for semantic relation (dependency) analysis such as (i) using RezoJDM for type-theoretic compositional semantics (Lafourcade et al., 2018); (ii) enhancing the quality of lexical-quantifier preference problem (Catta and Mirzapour, 2017); (iii) augmenting linguistic complexity measurement at syntactic level (Zou et al., 2022; Mirzapour et al., 2020) by using new inferred semantic relations.

References

Arora, S. (2020). A survey on graph neural networks for knowledge graph completion. *arXiv preprint arXiv:2007.12374*.

Balažević, I., C. Allen, and T. M. Hospedales (2019). Hypernetwork knowledge graph embeddings. In *International Conference on Artificial Neural Networks*, pp. 553–565. Springer.

Bollacker, K., C. Evans, P. Paritosh, T. Sturge, and J. Taylor (2008). Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pp. 1247–1250.

Bordes, A., N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko (2013a). Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems 26*.

Bordes, A., N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko (2013b). Translating Embeddings for Modeling Multi-relational Data. In C. J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems*, Volume 26. Curran Associates, Inc.

Cai, H., V. W. Zheng, and K. C.-C. Chang (2018). A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE Transactions on Knowledge and Data Engineering 30*(9), 1616–1637.

Catta, D. and M. Mirzapour (2017). Quantifier Scoping and Semantic Preferences. In *CONLI: Computing Natural Language Inference*, Montpellier, France.

Chen, Z., Y. Wang, B. Zhao, J. Cheng, X. Zhao, and Z. Duan (2020). Knowledge graph completion: A review. *IEEE Access 8*, 192435–192456.

Cousot, K., M. Mirzapour, and W. Ragheb (2019). Prediction of missing semantic relations in lexical-semantic network using random forest classifier.

Dettmers, T., P. Minervini, P. Stenetorp, and S. Riedel (2018). Convolutional 2d knowledge graph embeddings. In *Thirty-second AAAI conference on artificial intelligence*.

Duchi, J., E. Hazan, and Y. Singer (2011). Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research 12*(61), 2121–2159.

Ioffe, S. and C. Szegedy (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift.

Ji, G., S. He, L. Xu, K. Liu, and J. Zhao (2015). Knowledge graph embedding via dynamic mapping matrix. In *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers)*, pp. 687–696.

Lafourcade, M. (2007). Making people play for Lexical Acquisition with the JeuxDeMots prototype. In *SNLP'07: 7th International Symposium on Natural Language Processing*, Pattaya, Chonburi, Thailand, pp. 7.

Lafourcade, M., B. Mery, M. Mirzapour, R. Moot, and C. Retoré (2018). Collecting weighted coercions from crowd-sourced lexical data for compositional semantic analysis. In S. Arai, K. Kojima, K. Mineshima, D. Bekki, K. Satoh, and Y. Ohta (Eds.), *New Frontiers in Artificial Intelligence*, Cham, pp. 214–230. Springer International Publishing.

Lehmann, J., R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. Van Kleef, S. Auer, et al. (2015). Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web 6*(2), 167–195.

Lin, Y., Z. Liu, M. Sun, Y. Liu, and X. Zhu (2015). Learning entity and relation embeddings for knowledge graph completion. In *Twenty-ninth AAAI conference on artificial intelligence*.

Lu, H. and H. Hu (2020). Dense: An enhanced non-abelian group representation for knowledge graph embedding. *arXiv preprint arXiv:2008.04548*.

Mahesh, K. and S. Nirenburg (1997). *Knowledge-based systems for natural language processing*.

Miller, G. A. (1995). Wordnet: a lexical database for english. *Communications of the ACM 38*(11), 39–41.

Mirzapour, M., J.-P. Prost, and C. Retoré (2020). *Measuring Linguistic Complexity: Introducing a New Categorial Metric*, pp. 95–123. Cham: Springer International Publishing.

Mirzapour, M., W. Ragheb, M. J. Saeedizade, K. Cousot, H. Jacquenet, L. Carbon, and M. Lafourcade (2022). Introducing RezoJDM16k: a French Knowledge Graph DataSet for Link Prediction.

Nguyen, D. Q., T. D. Nguyen, D. Q. Nguyen, and D. Phung (2017). A novel embedding model for knowledge base completion based on convolutional neural network. *arXiv preprint arXiv:1712.02121*.

Nickel, M., V. Tresp, and H.-P. Kriegel (2011). A three-way model for collective learning on multi-relational data. In *Icml*.

Schlichtkrull, M., T. N. Kipf, P. Bloem, R. v. d. Berg, I. Titov, and M. Welling (2018). Modeling relational data with graph convolutional networks. In *European semantic web conference*, pp. 593–607. Springer.

Shang, C., Y. Tang, J. Huang, J. Bi, X. He, and B. Zhou (2019). End-to-end structure-aware convolutional networks for knowledge base completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Volume 33, pp. 3060–3067.

Simonyan, K. and A. Zisserman (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Socher, R., D. Chen, C. D. Manning, and A. Ng (2013). Reasoning with neural tensor networks for knowledge base completion. In *Advances in neural information processing systems*, pp. 926–934.

Srivastava, N., G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov (2014). Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res. 15*, 1929–1958.

Toutanova, K. and D. Chen (2015). Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd workshop on continuous vector space models and their compositionality*, pp. 57–66.

Trouillon, T., J. Welbl, S. Riedel, É. Gaussier, and G. Bouchard (2016). Complex embeddings for simple link prediction. In *International conference on machine learning*, pp. 2071–2080. PMLR.

Wang, M., L. Qiu, and X. Wang (2021). A survey on knowledge graph embeddings for link prediction. *Symmetry 13*(3), 485.

Wang, Z., J. Zhang, J. Feng, and Z. Chen (2014). Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Volume 28.

Yang, B., W.-t. Yih, X. He, J. Gao, and L. Deng (2014). Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*.

Zhang, S., H. Tong, J. Xu, and R. Maciejewski (2019). Graph convolutional networks: a comprehensive review. *Computational Social Networks 6*(1), 1–23.

Zou, L., M. Carl, M. Mirzapour, H. Jacquenet, and L. N. Vieira (2022). Ai-based syntactic complexity metrics and sight interpreting performance. In J.-H. Kim, M. Singh, J. Khan, U. S. Tiwary, M. Sur, and D. Singh (Eds.), *Intelligent Human Computer Interaction*, Cham, pp. 534–547. Springer International Publishing.

Résumé

In this paper, we present our ongoing research on a convolutional neural network architecture for knowledge graph completion task called Multi-layered ConvKB (MConvKB). Our new proposal is an extension to ConvKB with two main changes: (i) we added non-linear fully connected layers motivated by VGG architectures which acts as the last layer classifier, and (ii) we initialized the model with different pre-trained translational knowledge graph embeddings (TransE and TransD) to explore its effect in the model performance. Initialized model with TransD has shown better results. Experiments show that MConvKB can achieve significant results just by adding these two simple and yet effective modifications to the previous ConvKB model. To evaluated our approach we used RezoJDM16k, a relation prediction dataset for French. Our experiment show around 10% improvement compared to the original ConvKB model.

Une contribution du Text-mining à la connaissance du langage des cachalots

José Coch*, Lara Berkenbaum**, Olivier Adam***

*Dassault Systèmes
10, place de la Madeleine 75008 Paris France
jose.cochdiyacovo@3ds.com,
**23 avenue du manoir, 1640 Rhode-Saint-Genèse, Belgique
laraberkk@gmail.com,
***Institut d'Alembert Sorbonne Université
4 place Jussieu 75005 Paris France
olivier.adam@sorbonne-universite.fr

Résumé. Les cachalots communiquent avec des courtes séquences de clics successifs, appelées codas. Depuis une dizaine d'années, les interactions acoustiques entre cachalots sont enregistrées en particulier, à l'Ile Maurice. Partant des transcriptions de ces enregistrements, nous y avons ajouté des métadonnées sur l'âge, le sexe de chaque individu, et des relations familiales entre eux, puis nous avons étudié ces données avec Proxem Studio, un outil de Text-mining.

Nous présentons nos premiers résultats, qui tout d'abord montrent l'existence de signes composés (multi-codas). Nous observons aussi des corrélations appuyées entre certaines configurations (échanges vocaux entre mère et fils ou fille, conversations entre femelles adultes…) et certains signes, ainsi que des nombreuses corrélations entre des individus et des signes particuliers. Certains résultats entraîneraient des conséquences sur l'idée même de langage animal. Ces résultats nous encouragent pour la suite de nos travaux (dont nous donnons ici des pistes).

1 Introduction : la communication des cachalots

Depuis plus de 50 ans, ont été enregistrés des échanges vocaux ou des "conversations" entre les cachalots (Physeter macrocephalus), pouvant durer de quelques secondes, à une demi-heure (un cas d'un échange de 40 minutes a été observé en 2008 : Welch (2021)).

La communication des cachalots a la particularité d'être basée sur des clics. Il a été établi que leurs "messages" seraient basés sur de courtes séquences de plusieurs clics successifs, appelées "codas", de généralement trois à un peu plus d'une douzaine de clics relativement stéréotypées et récurrentes (Watkins et Schevill (1977), cité dans Weilgart et Whitehead (1993)).

Alors que les intervalles de temps entre 2 codas successives émises par le même cachalot peuvent être espacés dans le temps, les délais temporels entre les clics à l'intérieur d'une même coda sont, quant à eux, de durées beaucoup plus brèves. On parle de clics courts et clics moyens

ou longs faisant référence à la durée de l'intervalle inter-clic. Une méthode d'annotation de codas qui tient compte de cette caractéristique temporelle a d'ailleurs été définie.

Transcription simple. Les codas sont annotées en fonction des intervalles de temps entre clics. Lorsque ces derniers sont rapprochés, on les considère en paquet, et lorsqu'ils sont éloignés, on les compte individuellement. Le caractère "+" sert de lien. Ainsi, par exemple, la coda "3+1+1+2" représente une coda avec une suite de trois clics rapprochés (autrement dit séparés par des intervalles courts), ensuite un intervalle moyen, ensuite un clic, ensuite un autre intervalle moyen et un autre clic, et finalement un intervalle moyen et deux clics rapprochés (coda qui parfois est représentée par "/// / / //").

Transcription plus détaillée. Par ailleurs, les codas présentent des intervalles inter-clics plus ou moins réguliers. Il existe un autre mode de transcription rendant compte de ce dernier élément qui ajoute une lettre R lorsque les intervalles sont réguliers, D quand ils ne le sont pas, et L quand ils sont particulièrement brefs. Le corpus que nous avons utilisé pour le travail présenté ici n'utilise pas cette deuxième transcription plus détaillée mais la transcription simple décrite plus haut.

Les cachalots émettent d'autres types de sons, notamment les clics réguliers (utilisés pour l'écholocation), les bourdonnements ou "buzzes" (écholocation à courte distance Miller et al. (2004)), et les clangs (utilisés par les mâles adultes Weilgart et Whitehead (2011)). S'il semble évident que les codas servent à la communication, il n'est pas démontré que les bourdonnements et les autres sons ne jouent aucun rôle dans la communication. Il est par ailleurs possible que les clangs servent également à la communication, mais pour l'instant ils n'ont pas pu être étudiés comme les codas, les enregistrements acoustiques étant plus rares du fait que les adultes mâles sont plus isolés et éloignés des groupes sociaux qui sont principalement observés.

Les cachalots forment des groupes sociaux composés d'un ensemble de femelles adultes (jusqu'à plusieurs dizaines) accompagnées de juvéniles. On appelle ces groupes sociaux des "clans". Il est établi que chaque clan a son propre ensemble de codas. On parle souvent du "dialecte" de chaque clan.

Par exemple, comme nous allons le voir plus bas, dans le corpus d'échanges d'un clan de l'Ile Maurice (Océan Indien) que nous avons étudié, la coda la plus fréquente comporte 8 clics et est **2+1+1+1+1+1+1**, alors que dans un clan des Galapagos (Rendell et Whitehead (2004)), les codas les plus fréquentes sont les types **2+2** et **2+4**, par ailleurs absents du corpus du clan de l'Ile Maurice. Dans une publication un peu plus récente sur des cachalots des Caraïbes (Gero et al. (2016)), il a été montré que le type **1+1+3** (absent des Galapagos) y était un des plus fréquents. Encore plus récente, (Picozzi et al. (2022)) nous informe d'un clan dans le sanctuaire Pelagos (Méditerranée nord-occidentale), où le type **3+1** représente 73% du total.

En effet, les dialectes dépendent très fortement du clan et de la région : les cachalots des Galapagos n'ont pas exactement le même ensemble de codas que les cachalots des Caraïbes, de l'Atlantique, de la Méditerranée, et même d'autres zones du Pacifique. Citons finalement à ce propos (Hersh et al. (2022)) qui recense un nombre de clans dans le Pacifique et montre que chacun a des caractéristiques acoustiques/linguistiques bien spécifiques (et souvent une coda bien plus fréquente que les autres).

Le nombre de structures différentes de codas utilisées avec une fréquence significative par un clan se situe généralement entre 20 et 40.

Un ensemble de types de codas, qu'on pourrait assimiler à un *signaire*, est propre à un dialecte et donc à un clan. Il a été observé que ces "signaires" sont remarquablement stables

dans le temps et qu'ils semblent se transmettre par ligne maternelle (Rendell et al. (2012)).

Il a été remarqué une certaine richesse ou sophistication dans les "messages" des cachalots, qui sont enregistrés au cours d'interactions sociales entre plusieurs individus, même si à ce jour, il n'a pas été possible de décrire l'information sous-jacente qu'ils porteraient (Weilgart et Whitehead (1993)).

Le projet CETI (Andreas et al. (2021)) propose un ambitieux programme de travail pour "traduire" ou "déchiffrer" le langage des cétacés, et en premier lieu des cachalots. Ce projet observe les grandes avancées récentes en NLP et Machine Learning pour le langage humain, notamment les grands modèles de langage, et se propose d'utiliser ces outils pour le langage animal. Les approches envisagées exigeant des corpus extrêmement volumineux, une des premières étapes du CETI est donc de recueillir des corpus d'échanges de cachalots avec les meilleures techniques, et vu le volume visé, cela exige la mise au point de logiciels de traitement automatique des enregistrements acoustiques pour reconnaître les clics et les codas avec une grande précision. C'est par exemple l'objet de (Bermant et al. (2019)). A notre connaissance il n'est pas prévu de reconnaitre automatiquement l'individu émetteur de chaque coda, ni d'associer à chaque échange des informations sur le contexte. Il est pertinent de rappeler ici qu'une difficulté avec le langage des cachalots est que chaque clan a son propre dialecte ; il est donc très difficile d'obtenir des corpus cohérents très volumineux. Les grands modèles de langage demandent des grands corpus (des milliards de codas ?), mais des corpus cohérents. On dispose actuellement par exemple des corpus de dizaines de milliers de codas, mais qui correspondent à des dizaines de lieux différents, ce qui fait quelques milliers de codas par dialecte. Il s'agit donc d'un projet à long terme, qui à notre connaissance n'aborde pas encore l'étude du niveau syntaxico-sémantique.

Finalement citons un article récent (Bosshard et al. (2022)) qui, parlant de l'étude du langage des animaux en général (les cétacés ne sont pas particulièrement cités), souligne l'intérêt de l'étude des collocations dans le langage animal, autrement dit l'étude des suites de cris ou de signes qui semblent porter un sens spécifique.

2 Brève présentation de l'outil utilisé (Proxem Studio)

Proxem Studio est une suite logicielle multisource et multilingue de Dassault Systèmes dédiée à la collecte, l'analyse et la visualisation de données textuelles pour détecter les connaissances, les corrélations, les tendances et les signaux faibles, conçue et développée avec le but d'effectuer différents types d'analyse (d'opinion, de marché, de veille technologique, etc.).

Proxem Studio a permis de réaliser plusieurs centaines de projets d'application en analyse sémantique de données textuelles, notamment pour des grands comptes dans les domaines du transport, de l'industrie, de l'énergie, de la grande distribution, de la restauration, et de la haute technologie, mais aussi de nombreux *pure players* de l'analyse de l'opinion.

Depuis 2014, Proxem Studio intègre des techniques à base de réseaux de neurones, et a généralisé l'approche de type word embedding à l'apprentissage d'une ou même plusieurs langues (Coulmance et al. (2015)).

La philosophie de Proxem Studio est que l'intelligence artificielle soit au service des humains. Une réflexion ergonomique a été menée courant 2016 pour déterminer comment proposer à un utilisateur métier une application web simple d'utilisation, intégrant tous ces modules. Cette version de Proxem Studio a été lancée en mars 2017. Elle permet à cet utilisateur ayant le

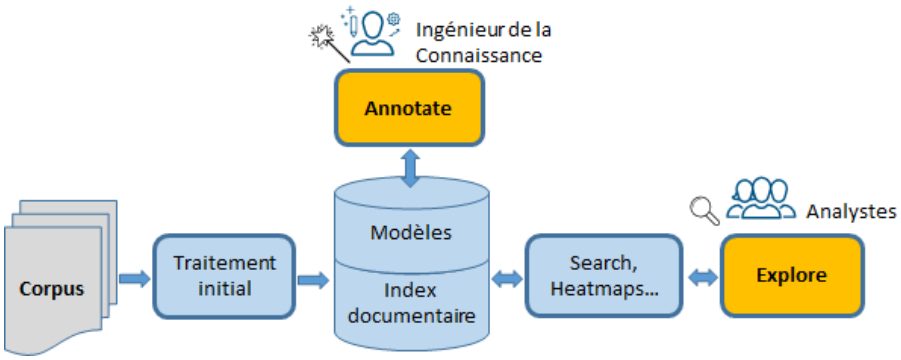


FIG. 1 – *Schéma des traitements et modules Annotate et Explore*

rôle d'Ingénieur de la Connaissance, qui n'a pas forcément de compétences en programmation ni *machine learning*, d'être autonome pour réaliser, de bout en bout, un projet comportant des tâches de web mining, text mining et data mining (Chaumartin (2017)).

La Figure 1 montre de manière simplifiée l'enchaînement des traitements et les principaux modules de Proxem Studio. Au moment de l'insertion d'un nouveau corpus, un **traitement initial** active un premier niveau d'analyse linguistique (quand un modèle de la langue est disponible) et réalise un certain nombre de calculs comme l'espace vectoriel des mots, la détection de termes (voir plus bas), et l'indexation documentaire. **Annotate** est un module pour l'Ingénieur de la Connaissance. Il permet d'effectuer une fouille de données textuelles (text-mining) et de construire un analyseur sémantique adapté à un corpus grâce à son moteur d'Intelligence Artificielle (avec un dictionnaire plus ou moins riche, et même sans dictionnaire du tout). Le module suggère au départ des candidats termes/concepts et des regroupements entre eux, et à tout moment des enrichissements des concepts en cours. Ces suggestions sont ensuite validées ou pas par l'ingénieur de connaissances en particulier en fonction des objectifs de chaque projet.

Annotate permet donc la détection et regroupement de termes mono et multi-mots, ainsi que l'extraction d'entités, de relations, la classification de séquences textuelles plus ou moins longues, l'analyse de sentiments, etc.

Explore est un autre module de Proxem Studio pour les utilisateurs ayant le rôle d'Analyste. Il permet d'exploiter l'analyseur sémantique construit avec Annotate, afin d'effectuer des statistiques, des histogrammes, des *treemaps*, des *tagclouds*, de retrouver des corrélations entre des éléments, notamment entre les termes et séquences du corpus et les métadonnées, d'étudier leurs évolutions dans le temps, etc.

Il existe deux spécialisations de Proxem Studio selon le type d'application :

— Proxem Insight, dédié à l'analyse d'opinions (typiquement dans le domaine de l'expérience client mais également l'expérience collaborateur et l'expérience citoyen),

— Proxem Knowledge, dédié à l'extraction d'information et la gestion des connaissances.

Proxem Knowledge permet en mode "Back-office", de produire et enrichir rapidement des référentiels (lexiques, ontologies, thésaurus, classifications…) métiers à partir des corpus du projet et/ou du domaine.

En mode plus "Front-office", Proxem Knowledge permet de réaliser des cartographies de marché, et des analyses assez approfondies de documents juridiques et techniques (contrats, actes notariés, appels d'offre, brevets, articles scientifiques, spécifications…) et produire des résultats actionnables.

De plus, Proxem Knowledge est utilisé pour réaliser des tâches de type indexation sémantique, clustering, anonymisation, et également *smart search* avec notamment la recherche *cross-language*.

2.1 Focus sur quelques caractéristiques techniques de Proxem Studio

Heatmaps (carte de chaleur, ou encore carte de fréquentation)

Pour étudier les éventuelles corrélations entre différents éléments et l'utilisation de telle ou telle coda, nous avons utilisé les heatmaps (voir par exemple Wilkinson et Friendly (2009)) produits par Proxem Studio.

Une carte de chaleur ou carte de fréquentation (en anglais "heatmap") permet de croiser différentes valeurs de deux éléments pour visualiser des sur-représentations statistiques dans une matrice où lesdites sur-représentations sont indiquées graphiquement par des couleurs plus foncées. Les valeurs numériques de la matrice de sur-représentation sont exportables.

Les différentes méthodes de calcul du score des heatmaps disponibles dans Proxem Studio sont les suivantes :

— Écart à la valeur théorique : analyse des sur-représentations, des signaux forts à l'intérieur du croisement ;
— Pourcentage de l'écart maximum : détection de signaux faibles, efficace sur des petits volumes ;
— Résidu du χ^2 : analyse de corrélations (mesure de la dépendance entre facettes) particulièrement pour les signaux faibles.

Sauf indication contraire, toute mention de heatmap dans le reste de ce document implique l'utilisation de la méthode d'écart à la valeur théorique.

Détection de termes

Il existe depuis déjà des dizaines d'années une préoccupation pour la suggestion automatique de termes et leur regroupement. Dans une première phase, l'approche technique utilisée pouvait avoir besoin d'un certain nombre de règles linguistiques, combinées éventuellement dans une certaine dose avec des considérations statistiques (par exemple Bourigault et Jacquemin (1999), Frantzi et al. (2000), Oliver et Vázquez (2015)). On peut remarquer dans les dernières décennies une tendance à l'augmentation relative de l'importance des aspects statistiques au détriment des règles linguistiques.

L'approche utilisée par Proxem Studio pour la détection de termes est basée sur l'Information mutuelle ponctuelle (ou "PMI" pour "Pointwise mutual information", voir Church et Hanks (1990)), sur sa version normalisée ("NPMI", Bouma (2009)). Le PMI mesure l'écart entre la probabilité de la co-occurrence de deux mots dans le corpus (dans notre cas, deux codas) et celle théorique calculée à partir des distributions individuelles, en supposant qu'il s'agisse d'évènements indépendants. L'ordre entre les mots est pris en compte : par exemple "pomme de terre" sera un candidat terme, "terre de pomme" beaucoup moins. L'avantage du

NPMI par rapport au PMI est que les valeurs du NPMI se trouvent entre -1 et 1, tandis que pour le PMI des valeurs absolues arbitrairement grandes sont possibles : le NPMI rend plus simple son traitement informatique. Lorsqu'un modèle de langue est disponible, la détection de termes utilise quelques informations linguistiques simples comme par exemple la catégorie morpho-syntaxique de chaque mot après analyse linguistique et prend en compte les mots vides de la langue.

Ce traitement est itéré en prenant compte les candidats termes à deux mots obtenus de la manière qu'on vient de décrire, permettant ainsi de détecter des candidats termes à plus de deux mots (exemple "pomme de terre nouvelle", etc).

Regroupement de termes

Un traitement de type word2vec (Mikolov et al. (2013), Bartunov et al. (2015)) appliqué à chaque corpus inséré dans Proxem Studio est réalisé afin d'obtenir un espace vectoriel des mots. Ceci rend possible un regroupement des termes en fonction de la proximité des contextes dans lesquels ils apparaissent, ce qui permet habituellement de détecter automatiquement des variantes graphiques ou morphologiques, des synonymes, et des termes proches. Ces suggestions automatiques ont pour conséquence une optimisation en termes de temps de projet, et également une bien meilleure exhaustivité dans la couverture linguistique des concepts.

Cette dernière technique est également utilisée pour l'enrichissement ponctuel des concepts : à partir d'un terme ou un concept donné, Annotate suggère des variantes ou synonymes qui pourront être regroupés avec le concept de départ.

Ces mécanismes permettent la mise en place assistée et optimisée de thésaurus mono ou multilingues (Perrais (2017)).

3 Données utilisées

Les données ont été collectées, entre 2013 et 2019 par l'association Longitude 181 dans le cadre du projet "Maubidick", porté par l'association Un Océan de Vie (de René Heuzey) et également par Longitude 181. Un groupe constitué de 17 femelles adultes et une dizaine de juvéniles a été filmé et enregistré acoustiquement lors de leurs interactions sociales de surface et sub-surface. Chaque cachalot a été identifié à partir des marques corporelles qui lui sont propres (Sarano et al. (2022)) et une analyse génétique a permis d'établir des liens de parenté entre les individus.

Pour l'étude présentée dans ce papier, nous avons utilisé une partie de ces données, en retenant finalement un corpus de 138 transcriptions d'échanges entre cachalots rattachés au clan IGT ("Irène Gueule Tordue") de l'Ile Maurice (Sarano et al. (2021), Sarano et al. (2022), Adam et al. (2020)), représentant un total de 1622 codas.

L'enregistrement n'étant pas stéréo il n'a pas été possible aux transcripteurs d'identifier l'émetteur de telle ou telle séquence sonore. Cependant, les participants à chaque conversation sont bien identifiés, ce qui est à souligner car c'est une information assez rare dans les corpus des communications des cachalots.

Le corpus se présente sous forme de fichier tabulé avec un certain nombre de métadonnées (individus participants, date, heure, etc.) et le "texte" (les transcriptions des codas). Voir Figure 2.

video	année	date	duree	nom indiv	début	codas	nbr_click
GOPR0867	2017	03/04/2017	00:00:56	Adelie_Eliot	00:00:21	2+1+1+1+1+1+1	8
GOPR0867	2017	03/04/2017	00:00:56	Adelie_Eliot	00:00:23	2+1+1+1+1+1	7
GOPR0867	2017	03/04/2017	00:00:56	Adelie_Eliot	00:00:27	2+1+1+1+1+1	7
GOPR0867	2017	03/04/2017	00:00:56	Adelie_Eliot	00:00:28	2+1+1+1+1+1+1+1	9
GOPR0867	2017	03/04/2017	00:00:56	Adelie_Eliot	00:00:30	2+1+1+1+1+1+1	8
GOPR0867	2017	03/04/2017	00:00:56	Adelie_Eliot	00:00:33	2+1+1+1+1+1	7
GOPR0867	2017	03/04/2017	00:00:56	Adelie_Eliot	00:00:37	3+1+1+1+1+1+1	9
GOPR0867	2017	03/04/2017	00:00:56	Adelie_Eliot	00:00:41	3+1+1+1+1+1+1	9
GOPR0867	2017	03/04/2017	00:00:56	Adelie_Eliot	00:00:46	3+1+1+1+1+1+1	9
GOPR0968	2017	06/04/2017	00:01:10	Vanessa_Zoe	00:00:40	5+1+1+1	8
GOPR0968	2017	06/04/2017	00:01:10	Vanessa_Zoe	00:00:41	2+1+1+1+1+1+4	11
GOPR0968	2017	06/04/2017	00:01:10	Vanessa_Zoe	00:00:44	2+1+1+1+1+1	7
GOPR0968	2017	06/04/2017	00:01:10	Vanessa_Zoe	00:00:51	2+1+1+1+1+1+1+1	9
GOPR0968	2017	06/04/2017	00:01:10	Vanessa_Zoe	00:00:56	2+1+1+1+1+1+1	8
GOPR0968	2017	06/04/2017	00:01:10	Vanessa_Zoe	00:01:08	5+1+1+1+1	9

FIG. 2 – *Extrait de la première version du tableau de données*

Nous avons également eu accès par ailleurs à des informations "démographiques" et généalogiques du clan d'IGT, ce qui nous a permis de rentrer certaines métadonnées complémentaires comme le sexe de chaque individu, sa tranche d'âge, et les relations familiales présentes dans chaque échange (mère-fils, mère-fille, frère-sœur, etc.) et ainsi obtenir un corpus enrichi avec ces nouvelles métadonnées (Sarano et al. (2021)).

3.1 Quelques statistiques simples sur les données

Sans avoir recours aux méthodes de Text-mining, on peut facilement observer que la très grande majorité des codas comportent 8 clics. Voir Figure 3.

Par ailleurs on observe environ 80 codas différentes dans les 138 échanges ; cependant seulement 29 d'entre elles ont une fréquence supérieure à 2 (voir Figure 4). La coda le plus fréquente **2+1+1+1+1+1+1**, a une fréquence d'un peu moins de 600.

Ajoutons que dans notre base de données, la très grande majorité des échanges concernent deux individus et que les cas des "monologues" existent mais sont rarissimes.

Finalement on peut indiquer que la moyenne des codas par conversation entre individus adultes est de 14, alors qu'elle tombe à 6 quand tous les participants sont jeunes ou immatures.

4 Méthodologie suivie

4.1 Préparation des données

Le format attendu par Proxem Studio veut que dans un fichier tabulé, chaque ligne contienne un verbatim ou une conversation (pas un mot par ligne, ou dans le cas des cachalots, pas une coda par ligne). Nous avons donc traité les données pour satisfaire cette exigence, et dans le même temps, nous avons ajouté à chaque ligne le nombre de codas de la conversation corres-

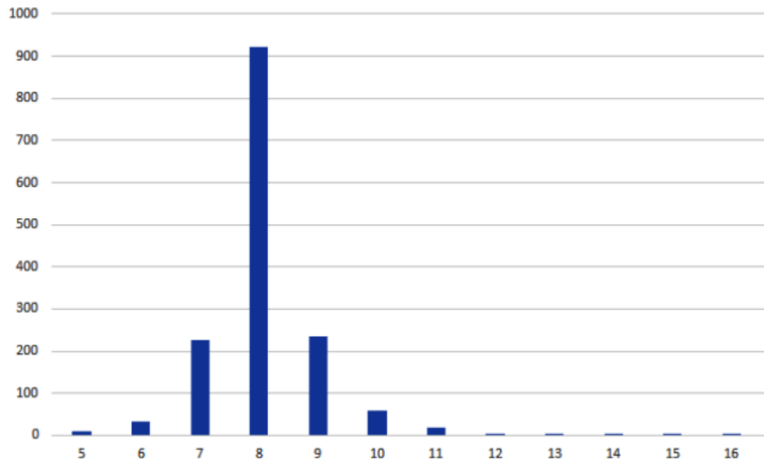


FIG. 3 – *Distribution des codas en fonction du nombre de clics*

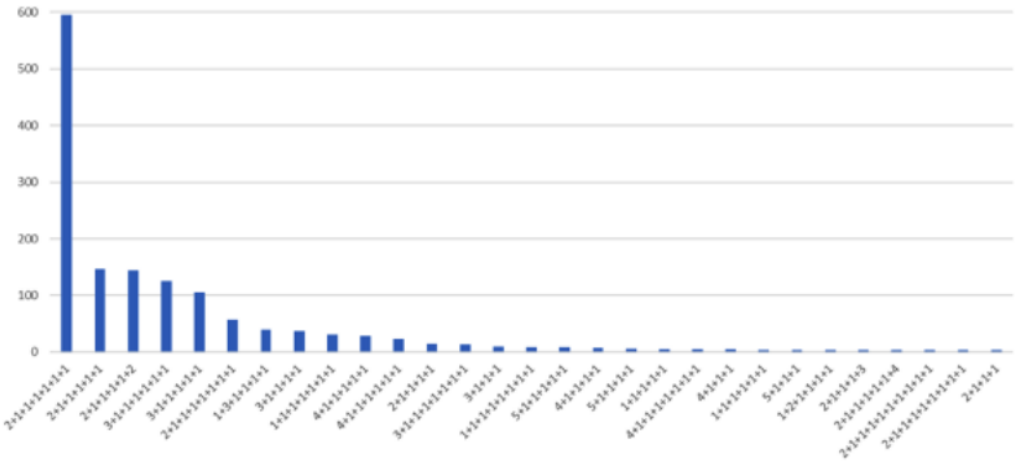


FIG. 4 – *Codas par fréquence dans le corpus (Top 29, freq>2)*

pondante. Nous avons procédé à d'autres normalisations (par exemple séparation et normalisation des noms des individus).

Important : profitant du fait que les individus intervenant dans les échanges ou "conversations" étaient identifiés, nous avons ajouté à chaque "conversation" des métadonnées venant des informations « démographiques » et généalogiques :

— sexe des individus,

— relations familiales entre les individus présents, notamment mère-fils ou mère-fille, sœurs ou frères, grand-mère et petit-fils ou petite-fille,
— tranches d'âge des individus.

Finalement nous avons itéré le test de chargement des données et la mise au point de celles-ci jusqu'à obtenir le chargement des données souhaité.

Ces itérations ont compris non seulement la correction et normalisation des données initiales, mais également une optimisation et enrichissement des métadonnées selon les résultats de chaque itération intermédiaire.

Par exemple, l'ajout de certaines relations familiales a rapidement semblé intéressante dès les premières itérations, alors que les informations sur l'âge ou plus exactement sur les tranches d'âge (adulte, immature, jeune) ont été ajoutées à la fin des itérations.

Paramétrage de l'outil

L'application des modules de Proxem Studio a permis de détecter de suites de codas apparaissant avec une fréquence significativement plus grande que s'il s'agissait d'une combinatoire au hasard. Nous avons donc ajouté ces "signes multi-codas" à la liste des codas pour former une liste de "candidats signes" à étudier et notamment à croiser avec des métadonnées, comme nous allons le voir dans la section suivante.

Analyse des résultats et rédaction d'un premier projet de rapport

Le croisement des signes (codas et multi-codas) avec un certain nombre de métadonnées, comme les relations familiales, les tranches d'âge mais aussi la longueur des conversations, nous a permis de détecter certaines corrélations fortes, et d'autres soit plus faibles, soit apparaissant comme fortes mais basées sur un nombre trop bas de cas. Nous ne présentons pas ces derniers cas ici, que nous considérons comme des pistes à confirmer ou infirmer une fois que nous disposerons d'un corpus plus large (voire section Future work - Suite des travaux).

Finalement nous avons rédigé un premier projet de rapport que nous avons présenté auprès de nos équipes et collègues, et qui nous a servi de base pour la rédaction du présent papier.

5 Présentation des résultats

5.1 Détection de multi-codas

L'outil détecte un certain nombre de suites de deux et parfois trois, quatre ou même cinq codas dont la fréquence est significativement supérieure à celle à laquelle on s'attendrait si la répartition des codas était au hasard. Dans les langues humaines, cette méthode bien connue (voir Church et Hanks (1990) cité plus haut) permet de détecter les termes, les mots composés et les idiomes.

Nous utilisons le mot "signe" pour faire référence aussi bien aux codas unitaires, qu'aux séquences multi-codas détectées par l'outil et validées par l'ingénieur des connaissances.

Parmi ces multi-codas il existe des répétitions de la même coda, et des suites de codas différentes.

Sans surprise, la coda la plus fréquente 2+1+1+1+1+1+1, donne lieu aux répétitions les plus fréquentes.

Autres bi-codas répétitives détectées :

— 2+1+1+1+1+2 2+1+1+1+1+2
— 2+1+1+1+1+1 2+1+1+1+1+1
— 3+1+1+1+1+1+1 3+1+1+1+1+1+1
— 1+3+1+1+1+1 1+3+1+1+1+1

Encore une fois, la coda 2+1+1+1+1+1+1 participe à 11 bi-codas sur un total de 22 bi-codas finalement retenues :

— 2+1+1+1+1+1 **2+1+1+1+1+1+1**
— **2+1+1+1+1+1+1** 2+1+1+1+1+1
— 2+1+1+1+1+2 **2+1+1+1+1+1+1**
— **2+1+1+1+1+1+1** 3+1+1+1+1+1+1
— 3+1+1+1+1+1+1 **2+1+1+1+1+1+1**
— **2+1+1+1+1+1+1** 2+1+1+1+1+2
— **2+1+1+1+1+1+1** 3+1+1+1+1+1
— 3+1+1+1+1+1 **2+1+1+1+1+1+1**
— 2+1+1+1+1+1+1+1 **2+1+1+1+1+1+1**
— 3+1+1+1+1 **2+1+1+1+1+1+1**
— 1+1+1+1+1+1+1 **2+1+1+1+1+1+1**

Il est intéressant de noter que ce dernier cas se présente dans 8% des conversations, tandis que la paire inverse (**2+1+1+1+1+1+1** 1+1+1+1+1+1+1) ne se présente que dans 0,7% des conversations. Cela semble démontrer que l'ordre entre les codas n'est pas aléatoire et surtout, que l'ordre entre les codas peut avoir une importance.

Finalement voici par exemple une bi-coda non répétitive sans la coda majoritaire :

— 2+1+1+1+1+2 2+1+1+1+1+1.

5.2 Corrélations de codas avec des métadonnées

Le croisement par les heatmaps (par écart à la valeur théorique, voir plus haut) des signes (codas et multi-codas) avec différents types de relations familiales (mère-fils ou mère-fille, frères-soeurs, soeurs-soeurs, grand-mères et petits-fils ou petites-filles, etc.) a mis en évidence la forte corrélation entre les conversations mère-fils/fille et certains signes. Des corrélations assez fortes semblent s'observer entre la relation soeur-soeur et certains signes, à confirmer ultérieurement avec des corpus plus conséquents, car pour le moment la volumétrie de ce type de conversations n'est pas assez significative.

C'est également le cas des conversations où tous les participants sont des mâles (immatures ou jeunes).

Signalons qu'il y a des corrélations très fortes entre les conversations où tous les membres sont adultes (femelles donc) et certains signes.

Finalement, des fortes corrélations sont observées entre certains individus et certains signes (un ou plusieurs par individu).

Mère-fils ou mère-fille

La heatmap suivante montre les corrélations entre les conversations entre une maman cachalot et son fils ou sa fille et les signes enregistrés.

La corrélation la plus forte entre la conversation d'une mère et son fils ou sa fille, est représentée par la coda **2+1+1+1+1+2** (effectif 23/45 avec une fréquence plus de deux fois supérieure que dans les autres conversations), suivie par **3+1+1+1+1+1+1** (effectif 26/45, fréquence

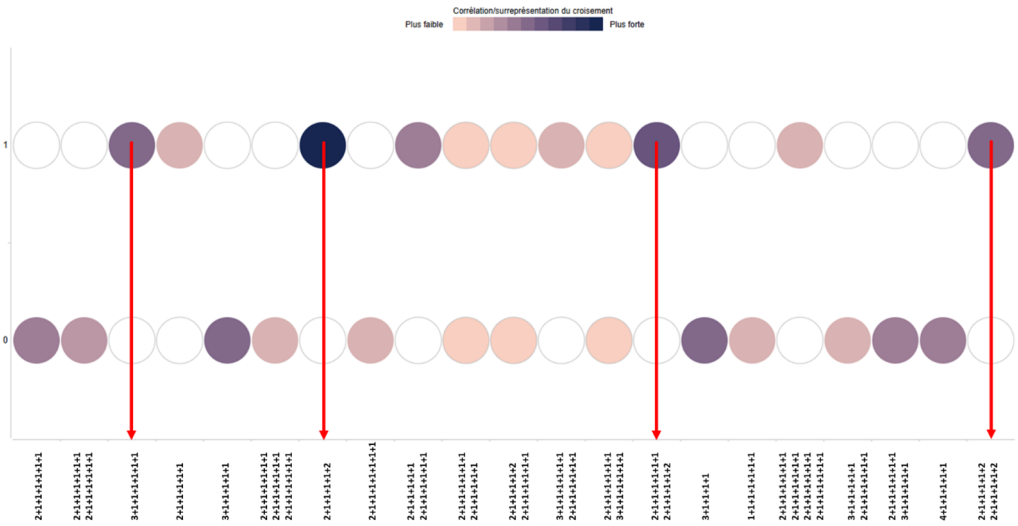


FIG. 5 – *Heatmap Mère-fils/fille vs Signes. Plus le cercle est foncé, plus la corrélation est forte*

27% supérieure), et les bi-codas **2+1+1+1+1+1+1 2+1+1+1+1+2** (effectif 14/45, fréquence 140% supérieure), et **2+1+1+1+1+2 2+1+1+1+1+2** (effectif 11/45, fréquence deux fois supérieure). A noter ici encore une fois que l'ordre entre codas semble avoir une importance car si **2+1+1+1+1+1+1 2+1+1+1+1+2** est sur-représentée dans les conversations mère-fils/fille, ce n'est pas du tout le cas de la paire inverse **2+1+1+1+1+2 2+1+1+1+1+1+1** (effectif 9/45, fréquence presque identique voire un petit peu inférieure).

Conversations par tranches d'âge

La Figure 6 montre les corrélations par tranche d'âge des participants et les signes enregistrés. Les trois tranches adulte, immature, jeune sont ici ramenées à deux : adultes et jeunes ou immatures, à cause de la faible fréquence des conversations entre jeunes seuls.

La corrélation la plus forte par rapport aux conversations où tous les individus sont adultes, est la bi-coda **2+1+1+1+1+2 2+1+1+1+1+1+1** (effectif 13/33, fréquence 175% supérieure). La coda la plus corrélée aux conversations mélangeant adultes et non-adultes est **3+1+1+1+1+1+1** (effectif 49/90, fréquence 63% supérieure). Nous avons déjà vu plus haut cette coda comme caractéristique des conversations mère-fils/fille.

Par individu

On peut observer dans la Figure 7, que certains individus semblent assez associés surtout à une coda spécifique, comme le cachalot nommé Ali à **4+1+1+1+1+1+1** (effectif 5/13, fréquence 4 fois supérieure), ou Eliot à **2+1+1+1+1+1** (effectif 11, fréquence 33% supérieure), mais le plus souvent les individus semblent être associés à plusieurs signes :

— Arthur à 2+1+1+1+1+1+1 2+1+1+1+1+1+1, 2+1+1+1+1+1+1 2+1+1+1+1+1+1 2+1+1+1+1+1+1, et 3+1+1+1+1+1+1 2+1+1+1+1+1+1 ;

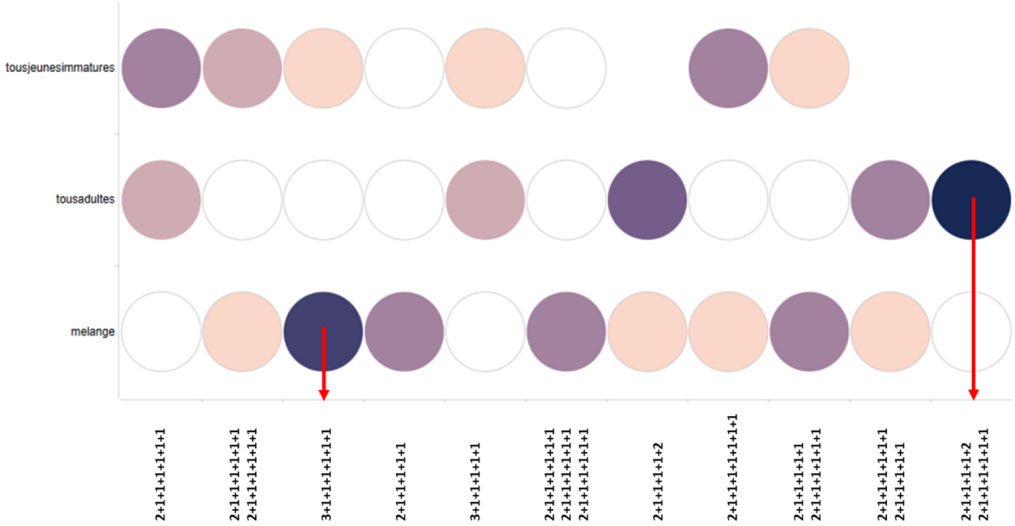


FIG. 6 – *Heatmap Tranche d'âge vs Signes*

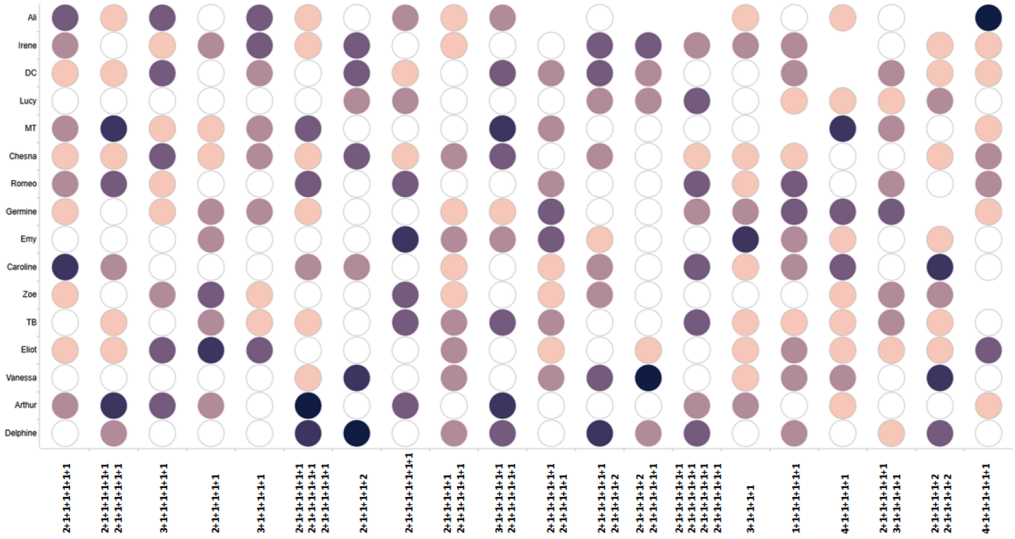


FIG. 7 – *Heatmap Individus vs Signes*

— Caroline à 2+1+1+1+1+1+1+1 et 2+1+1+1+1+1+2 2+1+1+1+1+1+2 ;
— Vanessa à 2+1+1+1+1+1+2 et 2+1+1+1+1+1+2 2+1+1+1+1+1+2 ;
— MT à 2+1+1+1+1+1+1+1 2+1+1+1+1+1+1+1, 3+1+1+1+1+1+1+1 2+1+1+1+1+1+1+1 et 4+1+1+1+1+1+1 ;

— Emy à 2+1+1+1+1+1+1+1 et 3+1+1+1+1 :
— Delphine à 2+1+1+1+1+2, 2+1+1+1+1+1+1 2+1+1+1+1+1+1 2+1+1+1+1+1+1 et 2+1+1+1+1+1+1 2+1+1+1+1+2 ;
— Zoe à 2+1+1+1+1+1 et 2+1+1+1+1+1+1+1.

Dans d'autres cas des corrélations existent mais elles sont un peu moins marquées :

— Lucy à 2+1+1+1+1+1+1 2+1+1+1+1+1+1 2+1+1+1+1+1+1 2+1+1+1+1+1+1 ;
— TB à 2+1+1+1+1+1+1+1 et 3+1+1+1+1+1+1 2+1+1+1+1+1+1 ;
— etc.

6 Discussion

Nous avons présenté brièvement la problématique de l'étude de la communication des cachalots, et décrit un travail réalisé à l'aide d'un outil de Text-mining initialement développé pour des projets commerciaux d'analyse d'opinion. L'outil pouvant être utilisé en mode "sans dictionnaire", nous avons pu l'utiliser pour étudier les transcriptions en codas des communications des cachalots du clan "IGT" de l'Ile Maurice.

En particulier, l'outil nous a permis de définir des signes multi-codas (un peu le correspondant cachalot des termes multi-mots ou mots composés du langage humain) qui arrivent significativement plus souvent que si c'était par simple probabilité mathématique.

De plus, nous avons profité de la disponibilité de données démographiques complémentaires pour les ajouter aux données initiales de façon à obtenir des métadonnées que nous avons pu croiser avec la liste de signes définie.

Par rapport aux autres projets que nous avons cités plus haut en fin de l'Introduction, nous pouvons souligner les suivants aspects originaux de notre travail :

— Nous utilisons des techniques qu'on peut assimiler au NLP et Machine Learning comme indiqué dans le programme du projet CETI. Cependant notre approche n'est pas du tout de viser la mise au point de grands modèles de langue de type GPT-3, mais au contraire d'appliquer des techniques notamment de Text-Mining pour avancer graduellement dans la connaissance des caractéristiques du langage des cachalots. A notre connaissance il n'existe pas dans le cadre du projet CETI des approches similaires à la nôtre.
— Nous n'avons pas connaissance de travaux externes aux auteurs du présent travail cherchant à associer des codas ou signes à des éléments contextuels comme dans nos métadonnées sur l'âge, le sexe et les relations familiales.
— La détection de multi-codas et l'étude de leur co-occurrence avec des éléments contextuels n'a jamais été faite à notre connaissance, même si elle peut être considérée à partir de l'article récent cité plus haut (Bosshard et al. (2022)).

Notre travail nous a permis de confirmer encore une fois que les clics et codas ne sont pas du tout produits au hasard, en particulier parce que le clan IGT n'utilise qu'un sous-ensemble assez petit des codas à priori possibles.

Notre travail semble montrer par ailleurs que l'ordre entre codas a son importance, par exemple comme nous l'avons vu plus haut pour les bi-codas associées aux conversations mère-fils/fille, et de manière plus abstraite pour certaines paires de codas qui dans un ordre donné, apparaissent plus de 10 fois plus souvent que dans l'ordre inverse.

Ce dernier point a une grande répercussion dans la connaissance générale du langage animal, car il va à l'encontre de l'idée assez répandue qui voudrait que le langage animal n'aurait pas de syntaxe et le sens serait donné par la simple superposition de composants (exemple souvent cité de la fameuse femelle gorille Koko, dominant 250 signes de la langue de signes américaine, pour qui "Koko want Banana" = "Banana Koko want" = "Banana want Koko", etc).

Notre travail nous a permis de montrer des corrélations entre certaines configurations (conversations familiales mère-fils/fille, et conversations entre femelles adultes) et certains signes.

Finalement nous avons pu montrer des corrélations entre des individus et certains signes.

D'autres corrélations, que nous appellerions des pistes, apparaissent clairement mais elles sont néanmoins à confirmer dans une étape ultérieure par l'étude de corpus complémentaires plus volumineux.

7 Future work - Suite des travaux

Les expéditions destinées à collecter des données audio-vidéos de cachalots continuent à l'île Maurice dans le cadre du projet Un Océan De Vie en collaboration avec l'association Longitude 181. L'objectif étant d'approfondir les recherches et connaissances sur ces mammifères marins.

Un projet de thèse de doctorat est par ailleurs en cours, candidaté par la biologiste Lara Berkenbaum, dirigé par le bio acousticien Olivier Adam ainsi que l'équipe de l'océanographe François Sarano, et accompagné sur certains aspects par le mathématicien et linguiste José Coch. Cette thèse ainsi que d'autres recherches en cours visent principalement à décrire les interactions entre individus sur base de l'interprétation de comportements socio-acoustiques. Ces travaux visent à mieux comprendre les communications vocales et à mettre en évidence les liens entre sons et comportements spécifiques à l'aide d'outils permettant l'identification du ou des locuteurs.

Un répertoire comportemental a été décrit dans (Berkenbaum (2021) et Adam et al. (2020)). Il met en évidence différents comportements sociaux observés lors d'interactions entre individus et classés en plusieurs catégories (contact, déplacement, exploration, etc.) de premier niveau, et une petite vingtaine de sous-catégories.

Nous espérons que les méthodes de Text-mining comme celles décrites ici, en exploitant des corpus plus volumineux et plus précis, et en enrichissant les métadonnées en ajoutant des informations sur les comportements sociaux, nous permettront d'approfondir la connaissance du sophistiqué langage des cachalots.

8 Remerciements

Merci au Docteur François Sarano (avec l'association Longitude 181) et à René Heuzey (avec sa société Label Bleu Production et son association Un Océan de Vie) pour le travail extraordinaire qu'ils mènent depuis plus de 10 ans sur les cachalots de Maurice et qui nous ont fourni les enregistrements audio-vidéos. Ce travail n'aurait pas pu exister sans eux. Merci également à Jocelyn Coulmance, Directeur de Technologie NLP chez Dassault Systèmes, pour son aide dans la description de Proxem Studio.

Références

Adam, O., A. Yernaux, M. Sauvêtre, J. Ngosso, G. Nuel, M. Haffner-Trinh, R. Troussier, Z.-L. Guillerm, L. Picon, L. Barluet, J. Macky, L. Barluet de Beauchesne, V. Kuhn, F. Delfour, V. Sarano, H. Vitry, A. Preud'homme, R. Heuzey, J.-L. Jung, et F. Sarano (2020). Study of behaviours and emitted codas during sperm whale social interactions. *e-Forum Acusticum 2020, Dec 2020, Lyon, France. pp.3225-3227*.

Andreas, J., G. Begus, M. Bronstein, R. Diamant, D. Delaney, S. Gero, S. Goldwasser, D. Gruber, S. Haas, P. Malkin, R. Payne, G. Petri, D. Rus, P. Sharma, P. Tønnesen, A. Torralba, D. Vogt, et R. Wood (2021). Cetacean translation initiative : a roadmap to deciphering the communication of sperm whales. *arXiv* (2104.08614).

Bartunov, S., D. Kondrashkin, A. Osokin, et D. Vetrov (2015). Breaking sticks and ambiguities with adaptive skip-gram. *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, PMLR 51*, 130–138.

Berkenbaum, L. (2021). Étude étho-acoustique d'un groupe social de cachalots (physeter macrocephalus) résident de l'Île maurice (océan indien). *Université de Liège, Liège, Belgique. Unpublished master's thesis. URL : matheo.uliege.be/handle/2268.2/12623*.

Bermant, P., M. Bronstein, R. Wood, S. Gero, et D. Gruber (2019). Deep machine learning techniques for the detection and classification of sperm whale bioacoustics. *Scientific Reports 9*, 1–10.

Bosshard, A., M. Leroux, N. Lester, B. Bickel, S. Stoll, et S. Townsend (2022). From collocations to call-ocations : using linguistic methods to quantify animal call combinations. *Behavioral Ecology and Sociobiology ; Heidelberg 76*.

Bouma, G. (2009). Normalized (pointwise) mutual information in collocation extraction. In *From Form to Meaning : Processing Texts Automatically, Proceedings of the Biennial GSCL Conference 2009*, Tubingen, Allemagne, pp. 31–40.

Bourigault, D. et C. Jacquemin (1999). Term extraction + term clustering : An integrated platform for computer-aided terminology. In *Ninth Conference of the European Chapter of the Association for Computational Linguistics*, Bergen, Norway, pp. 15–22. Association for Computational Linguistics.

Chaumartin, F. (2017). Proxem studio : la plate-forme d'analyse sémantique qui transforme l'utilisateur métier en text scientist. In *Actes des 24 Conférence sur le Traitement Automatique des Langues Naturelles. Volume 3. Demonstrations*, Orléans, France, pp. 35–36. ATALA.

Church, K. W. et P. Hanks (1990). Word association norms, mutual information, and lexicography. *Computational Linguistics 16*(1), 22–29.

Coulmance, J., J.-M. Marty, G. Wenzek, et A. Benhalloum (2015). Trans-gram, fast cross-lingual word-embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal, pp. 1109–1113. Association for Computational Linguistics.

Frantzi, K., S. Ananiadou, et H. Mima (2000). Automatic recognition of multi-word terms : The c-value/ nc-value method. In *Int. J. on Digital Libraries*, Volume 3, pp. 115–130.

Gero, S., H. Whitehead, et L. Rendell (2016). Individual, unit and vocal clan level identity cues in sperm whale codas. *Royal Society Open Science 3*, 150372.

Hersh, T., S. Gero, L. Rendell, M. Cantor, L. Weilgart, M. Amano, S. Dawson, E. Slooten, C. Johnson, I. Kerr, R. Payne, A. Rogan, R. Antunes, O. Andrews, E. Ferguson, C. Hom-Weaver, T. Norris, Y. Barkley, K. Merkens, et H. Whitehead (2022). Evidence from sperm whale clans of symbolic marking in non-human cultures. *Proceedings of the National Academy of Sciences 119*.

Mikolov, T., K. Chen, G. Corrado, et J. Dean (2013). Efficient estimation of word representations in vector space. *Proceedings of Workshop at ICLR 2013*.

Miller, P., M. Johnson, et P. Tyack (2004). Sperm whale behaviour indicates use of echolocation click buzzes 'creaks' in prey capture. *Proceedings. Biological sciences / The Royal Society 271*, 2239–47.

Oliver, A. et M. Vázquez (2015). Tbxtools : A free, fast and flexible tool for automatic terminology extraction. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*.

Perrais, T. (2017). Construction de thésaurus assistée par machine learning. In *Journée commune AFIA-ARIA*, URL : ia-ri.sciencesconf.org/data/Perrais_proxem.pdf.

Picozzi, J., S. Panigada, S. Airoldi, N. Pierantonio, et G. Pavan (2022). Sperm whale coda vocal repertoire in the pelagos sanctuary, north-western mediterranean sea. In *33rd Conf. European Cetacean Society*, pp. 171.

Rendell, L., S. Mesnick, M. Dalebout, J. Burtenshaw, et H. Whitehead (2012). Can genetic differences explain vocal dialect variation in sperm whales, physeter macrocephalus ? *Behavior genetics 42*, 332–43.

Rendell, L. et H. Whitehead (2004). Do sperm whales share coda vocalizations ? insights into coda usage from acoustic size measurement. *Animal Behaviour 67*, 865–874.

Sarano, F., J. Girardet, V. Sarano, H. Vitry, A. Preud'homme, R. Heuzey, A. Garcia Cegarra, B. Madon, F. Delfour, H. Glotin, O. Adam, et J.-L. Jung (2021). Kin relationships in cultural species of the marine realm : case study of a matrilineal social group of sperm whales off mauritius island, indian ocean. *Royal Society Open Science 8*.

Sarano, V., F. Sarano, J. Girardet, A. Preud'homme, H. Vitry, R. Heuzey, M. Sarano, F. Delfour, H. Glotin, O. Adam, B. Madon, et J.-L. Jung (2022). Underwater photo-identification of sperm whales (physeter macrocephalus) off mauritius. *Marine Biology Research 18*(1-2), 131–146.

Watkins, W. A. et W. E. Schevill (1977). Sperm whale codas. *The Journal of the Acoustical Society of America 62*(6), 1485–1490.

Weilgart, L. et H. Whitehead (2011). Distinctive vocalizations from mature sperm whales (physeter macrocephalus). *Canadian Journal of Zoology 66*, 1931–1937.

Weilgart, L. S. et H. Whitehead (1993). Coda communication by sperm whales (physeter macrocephalus) off the galápagos islands. *Canadian Journal of Zoology 71*, 744–752.

Welch, C. (2021). Groundbreaking effort launched to decode whale language. *National Geographic. URL : www.nationalgeographic.com/animals/article/scientists-plan-to-use-ai-to-try-to-decode-the-language-of-whales*.

Wilkinson, L. et M. Friendly (2009). The history of the cluster heat map. *The American Statistician 63*, 179–184.

Summary

Sperm whales communicate with small sequences of successive clicks, called codas. For the past ten years, 'conversations' between sperm whales have been recorded, particularly in Mauritius. Starting from the transcriptions of these recordings, we added metadata on the age, sex of each individual, and family relationships between them, then studied these data with Proxem Studio, a text-mining tool.

We present our first results, which primarily show the existence of compound signs (multi-codas). We also observe strong correlations between certain configurations (conversations between mother and son or daughter, conversations between adult females, etc.) and certain signs, as well as numerous correlations between individuals and particular signs. Some of the results would have consequences for the very idea of animal language. These results encourage us for the continuation of our work (of which we give some hints here).

Traitement automatique de la coréfénce dans les textes économiques : un exemple d'application chez ReportLinker

Marilyne Latour* Paul Moncuquet**

ReportLinker, 21 Quai Antoine Riboud, 69002 Lyon, France
*mla@reportlinker.com,**pmo@reporlinker.com
http://www.reportlinker.com

Résumé. Cet article présente un retour d'expérience sur le phénomène de résolution de coréférence. L'expérience consiste à observer le corpus à partir de phrases issues de dépêches d'actualité en économie afin d'interpréter les liaisons entre deux expressions référentielles qui renvoient à une même entité. Nous avons observé les cas de coréférence dans un type de texte particulier : les dépêches d'actualité en anglais évoquant l'actualité économique et financière. Nous avons étudié les cas de coréférence dans deux situations : entre un nom propre et un syntagme nominal (relations de périphrase) *« Globus Medical, Inc. »* et *« The medical device company »*), et entre un nom propre et un pronom (relations anaphoriques) *« Salona Global Medical Device Corporation »* et *« it »*). Notre expérience à ensuite consisté à évaluer s'il était possible de lier une coréférence à son antécédent à partir de deux relations logiques : en le liant [1] à la phrase précédente [2] au titre de la dépêche d'actualité.

1 Introduction

La prise en compte des relations de coréférence est un domaine en développement du traitement automatique du langage (TAL). Elle correspond au fait de pouvoir prendre en compte l'ensemble des expressions référentielles qui désignent le même référent dans le discours. Ces expressions référentielles sont alors coréférentes et permettent de récupérer davantage d'informations sémantiques et une meilleure prise en compte du sens du texte. La prise en compte de la coréférence en corpus offre des opportunités de performance sur les systèmes de recherche d'informations (SRI) comme la qualité des informations sémantiques obtenues.

2 Etat de l'art

La coréférence est la relation entre plusieurs expressions référentielles qui désignent le même référent (Delaborde, 2021). En anglais, on parle de « coreference chain », « coreferential chain » ou simplement de « coreferences ». En français, elles sont nommées « chaîne de référence » par de nombreux linguistes francophones (Gobert et Fabre, 2017) ; (Landragin et Oberle, 2018). Plusieurs travaux en linguistique (morphologie, syntaxe, rôles discoursifs et sémantique) portent sur les entités du discours et donnent des indications sur les critères qui

peuvent aider les systèmes de traitement automatique à prédire si une expression référentielle sera reprise (coréférente) ou non (singleton). (Recasens et al., 2013) ont pris en compte ces caractéristiques pour les intégrer à un modèle capable de distinguer les singletons des maillons coréférents. D'autres travaux se sont aussi penchés sur la détection des chaînes en fonction du type de texte (Boudreau, 2004). Dans ce domaine, l'étude de la coréférence se fait de manière globale, prenant en compte les chaînes de coréférence à partir de deux ou plusieurs expressions référentielles, alors appelés «maillons» au sein d'un même texte. C'est dans cette perspective que se situe notre expérience.

3 Contexte applicatif

ReportLinker [1] est un moteur de recherche qui fournit un accès direct et organisé aux documents économiques édités par 200 000 sources d'information différentes sous formes de rapports de marché, de statistiques, d'études de marché ou encore de profils d'entreprises. Il fournit également mensuellement 4 à 6 millions de dépêches d'actualité en langue anglaise, réparties sur 18 000 journaux ou sites web. Nous nous situons pour cette étude exclusivement sur l'étude des dépêches d'actualité pour deux raisons principales : la fraîcheur et la quantité des informations extraites.

Au sein du moteur de recherche, la phrase est l'unité d'information privilégiée. Plus spécifiquement, les phrases qui présentent des *données factuelles* par exemple des données qui sont de l'ordre de l'observable comme dans l'exemple ci dessous :

> « Hifyre TM launched the Spark Perks TM members program providing benefits such as Fastlane checkout, exclusive deals and access to member-only events. »

Nous appelons ce type de phrases les phrases « cibles ». Notre objectif est alors d'extraire, étiqueter et catégoriser automatiquement au niveau de ces phrases cibles les données suivantes : secteur d'activité, localisation, données chiffrées, noms de société ou de produits, etc. Chaque phrase -prise individuellement- ne comporte pas un degré informationnel aussi riche ; il est donc intéressant d'un point de vue de la catégorisation de pouvoir faire « hériter » des informations d'une phrase à une autre. Dans le présent travail, nous nous sommes intéressés aux noms des sociétés absents des phrases cibles, représentés sous formes de syntagme nominal (« The medical device company », «This firm »), ou de pronom (« it »). La résolution de coréférence semble être une alternative intéressante pour le traitement de ce genre de données ; elle pourrait ainsi permettre la catégorisation des phrases cibles qui, pour l'instant, contiennent des données manquantes lors du module d'analyse intégré au moteur de recherche (Figure 1) et qui ne sont donc pas retenues lors de l'analyse sémantique.

4 Expérience

L'expérience a consisté à extraire 10 000 phrases provenant de dépêches d'actualité ayant été catégorisées par le moteur de recherche comme étant de type « société » et sous-catégorisées

1. www.reportlinker.com

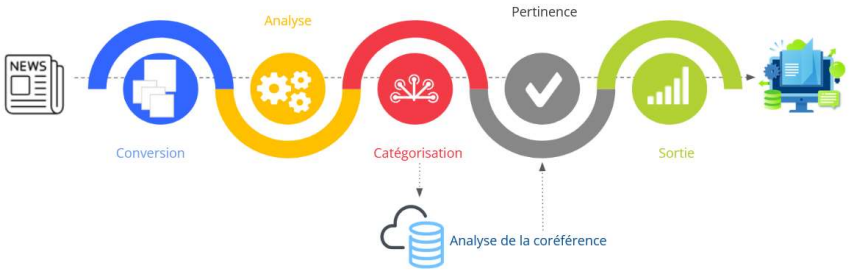


FIG. 1 – *Analyse sémantique de la dépêche d'actualité par le biais de modules sémantiques intégrés dans le moteur de recherche*

parmi les cinq classes suivantes : fusions et acquisitions, données de vente, données financières, lancements de produit ou descriptions d'activité d'une société [2].

La deuxième étape a été de repérer les expressions référentielles (*e.g.* « The company », « The firm », « it ») dans les phrases qui ne mentionnaient pas explicitement les noms de société. Ces expressions ont été relevées humainement. Nous avons alors testé l'hypothèse suivante : est-il possible de prédire le référent (par exemple le nom de la société) dans les cas de relations de périphrases et les cas de relations d'anaphores en analysant le nom de la société mentionnée dans la phrase précédente ? En effet, la coréférence s'opère en grande majorité d'une phrase à un autre en raison de la proximité textuelle des maillons coréférents.

La troisième étape a été de vérifier si la distance « inter-maillonnaire » (Ariel, 1990) joue un impact sur la fiabilité de la coréférence. Pour cela, nous avons testé si les expressions référentielles dans les phrases issues de ce corpus pouvait être reliées aux référents mentionnés dans les titres. En effet, en raison de la forte valeur identifiante des noms de société dans ces types de dépêches (le nom de la société est souvent repris comme par exemple dans la présentation des résultats financiers), il est intéressant de suivre le «devenir discursif» (Schnedecker, 2019) d'un référent au fil du discours narratif pour étudier la répartition des maillons dans ce genre de texte.

4.1 Constitution du corpus

Nous avons constitué un corpus de 10 000 phrases cibles traitant de fusions et acquisitions, données de vente, données financières, lancements de produit et descriptions d'activité d'une société avec une répartition équivalente pour chacune des classes (2 000 chaque). Pour chacune de ces 10 000 phrases cibles, nous avons extraits trois blocs de texte :

— la phrase cible elle-même,
— la phrase précédant à la phrase cible,

2. Cette analyse s'opère lors du module Catégorisation dans la chaîne de traitement sémantique du moteur de recherche (troisième module de la Figure 1).

— le titre de la dépêche d'actualité.

4.2 Annotation du corpus

Afin que le schéma d'annotation reste simple, nous avons compilé ces informations dans un logiciel tableur avec un accès direct à la dépêche d'actualité via un lien URL. Un cadre théorique précis a été répertorié dans le guideline d'annotation et fourni à quatre annotateurs spécialistes en économie. La phase d'annotation s'est organisée en plusieurs temps :

1. **Annotation [Phrase_cible]** : si la phrase contenait bien le nom de la société dans son intégralité (*e.g.* « Globus Medical Inc. ») alors les annotateurs indiquaient que la phrase cible était compréhensible et passaient à la suivante.
   Si la phrase cible contenait un syntagme nominal faisant référence à une société comme « This company » ou « The medical device company » ou le pronom « it » alors les annotateurs les annotaient comme non compréhensibles et devaient annoter deux champs supplémentaires :
2. **Annotation [Phrase_Precedente]** : si la phrase précédente ne faisait mention qu'à une seule société et si cette société était bien le référent du maillon de la phrase suivante, alors les annotateurs confirmaient la chaîne de coréférence [Phrase_Precedente] <–> [Phrase_cible],
3. **Annotation [Titre_Depeche]** : si le titre ne faisait mention qu'à une seule société et si cette société était bien le référent du maillon de la phrase cible, alors les annotateurs confirmaient la chaîne de coréférence [Titre_Depeche] <–> [Phrase_cible].

C'était également le rôle des annotateurs de renseigner le référent de chaque expression référentielle par exemple le nom de la société concernée dans la phrase cible. Un accord inter-annotateur a été mis en place afin de s'assurer de la conformité et de l'homogénéité des annotations. Le contenu a été réduit à l'essentiel : *'1'* si il y a avait accord, et *'0'* si il y avait désaccord. Nous avons obtenu un kappa score de 0.80, qui est une valeur acceptée comme gage de bonne fiabilité des données annotées (Dany et al., 2019). Cette méthodologie d'annotation nous a permis d'avoir un corpus facilement exploitable. Les annotations réalisées pour le projet ont pu être analysées de manière statistique.

5 Analyse

L'analyse des résultats nous a permis de mettre en évidence plusieurs éléments. Le premier porte sur les phrases cibles non compréhensibles.

Exemple 1 : Phrases cibles non compréhensibles

[Phrase_cible]*« The Swiss-based company is launching a slew of new products aimed especially at people working from home. »*

On voit à partir de l'exemple 1 que la phrase cible est non compréhensible par l'élément « company » qui demande donc à être précisé dans son contexte.

A contrario, les phrases compréhensibles ne nécessitent pas d'autres éléments de contextuation.

Exemple 2 : Phrases cibles compréhensibles

[Phrase_cible] *« Provident Healthcare Partners ("Provident"), a leading healthcare investment banking firm, announced that it has advised Texan Eye and Eye LASIK Austin in their partnership with Comprehensive EyeCare Partners ("CompEye"). »*

Dans notre corpus, 3 591 phrases (soit 36%) pourraient bénéficier de l'apport de la résolution de coréférence pour enrichir la base de données du moteur de recherche 1.

	Nombre de phrases	Pourcentage
Phrases cibles compréhensibles	6409	64%
Phrases cibles non compréhensibles	3591	36%

TAB. 1 – *Répartition du nombre de phrases cibles compréhensibles et non compréhensibles parmi le corpus de 10 000 phrases*

A partir de ce corpus de phrases non compréhensibles, nous avons testé quatre échantillons :

— **Modèle 1** : un seul même et unique nom de société présent à la fois dans la phrase précédente et dans le titre.
— **Modèle 2** : un seul nom de société présent uniquement dans la phrase précédente,
— **Modèle 3** : un seul nom de société présent uniquement dans le titre,
— **Modèle 4** : pas de mention de nom de société, ni dans la phrase précédente, ni dans le titre ou noms de sociétés différents.

Exemple Modèle 1

En reprenant l'exemple 1 dont [Phrase_cible] n'était pas compréhensible sans élément de contexte (« The Swiss-based company »), la société est à la fois mentionnée dans la [Phrase_Precedente] et [Titre_Depeche] : « Logitech ».

— **[Phrase_cible]** *« The Swiss-based company is launching a slew of new products aimed especially at people working from home. »*
— **[Phrase_Precedente]** *« Working from home is something that Logitech is taking very seriously. »*
— **[Titre_Depeche]** *« Logitech Introduces All-In-One Dock to Declutter the Desktop and Make Joining Meetings Easy »*

Exemple Modèle 2

Dans cet exemple, il manque un élément de contexte par rapport à la [Phrase_cible] (« The tech company ») ; la société est citée dans la [Phrase_Precedente] mais pas dans le titre.

— **[Phrase_cible]** *« The tech company took a major step toward that goal earlier this year with its acquisition of Hyperconnect in a cash-and-stock transaction valued at $1.725 billion. »*

— **[Phrase_Precedente]** *« But Match Group is also looking to expand its offerings beyond dating. »*
— **[Titre_Depeche]** *« 2 Stocks That Could Turn $200,000 Into $1,000,000 in 10 Years. »*

Exemple Modèle 3

La [Phrase_cible] présente là aussi un élément manquant (« The firm ») tout comme dans la [Phrase_Precedente] (« The company »). Seul un nom de société est présent dans le [Titre_Depeche].

— **[Phrase_cible]** *« The firm had revenue of C$308.26 million during the quarter, compared to the consensus estimate of C$305.00 million. »*
— **[Phrase_Precedente]** *« The company reported C$0.51 earnings per share for the quarter. »*
— **[Titre_Depeche]** *« High Liner Foods (TSE :HLF) Shares Cross Above 200 Day Moving Average of $0.00 »*

Exemple Modèle 4

Dans cet exemple, la [Phrase_cible] a besoin d'un élément de contexte sur de nom de société « The company » qui n'est fourni ni dans la [Phrase_Precedente] (pronom « it »), ni dans le [Titre_Depeche] (absence d'expressions référentielles).

— **[Phrase_cible]** *« The company is seeing strong same-store sales growth, up 51% in the first quarter of 2021 over Q1 2020. »*
— **[Phrase_Precedente]** *« It sells all of the equipment and supplies for growing, including climate controls, lighting, soils, fertilizer, and additives. »*
— **[Titre_Depeche]** *« Looking for a U.S. Cannabis Stock ? This Might Be the One »*

Résultats

Le modèle 1 est le plus sûr car la chaîne de coréférence peut se baser sur la répétition du même nom de la société à partir de deux blocs de texte stratégiques : la phrase qui la précède la phrase cible et le titre. D'après le tableau 2, ce modèle pourrait permettre de traiter la coréférence en moyenne dans un peu plus de 25% des cas (pouvant aller jusqu'à 65% des cas pour certaines classes comme les phrases issues de la classe « Données financières »). Le modèle 3 est intéressant par sa volumétrie : il permettrait d'apporter du contexte dans 50% des cas (69% des cas dans le cas des phrases issues de la classe « Données de vente »). Le modèle 2 a le même problème de fiabilité que le modèle 3 et ne concernerait que 5% des cas. Le modèle 4 n'apporte pas de contexte supplémentaire utile pour le traitement de la coréférence.

Nous avons également testé la précision de ces modèles : à partir des renseignements des annotateurs humains sur le référent de chaque expression référentielle (par exemple le nom de la société concernée dans la phrase cible), nous avons pu comparer les résultats avec les entités nommées qui avaient été extraites automatiquement de [Phrase_Precedente] et [Titre_Depeche] par notre analyseur sémantique. Les résultats sont présentés dans le tableau 3.

Le tableau 3 met en valeur la précision et la fiabilité de cette proposition de traitement de la coréférence par héritage du nom de la société provenant soit de la phrase précédente et du titre dans 98% des cas (modèle 1), soit de la phrase précédente uniquement dans 99% des

	Modèle 1	Modèle 2	Modèle 3	Modèle 4
Fusions et acquisitions	29%	6%	58%	7%
Données de vente	23%	2%	69%	6%
Données financières	65%	1%	33%	1%
Lancements de produits	25%	10%	47%	18%
Descriptions d'activités	40%	7%	44%	9%

TAB. 2 – *Répartition du nombre phrases « cibles » compréhensibles parmi le corpus de 10 000 phrases*

	Modèle 1	Modèle 2	Modèle 3	Modèle 4
Précision	98%	99%	96%	–

TAB. 3 – *Précision de la coréférence en fonction des 4 modèles*

cas (modèle 2), soit du titre uniquement uniquement dans 96% des cas (modèle 3). Ces bons chiffres peuvent éventuellement s'expliquer par le choix du corpus : des dépêches d'actualité sur la finance avec des typologies de classes spécifiques aux sociétés. La principale source d'erreurs -notamment du modèle 3- est lorsque plusieurs noms de sociétés sont cités dans le [Titre_Depeche], apportant de la confusion lors de l'interprétation des expressions référentielles. Même cas de figure pour le modèle 2 avec la [Phrase_Precedente] mais qui est plus rare. Il semblerait donc que la distance « inter-maillonnaire » joue un impact sur la fiabilité de la coréférence.

Exemple d'erreurs :

— **[Phrase_cible]** *« The company recorded slow year-on-year growth for hair products, and the sales under hair saloon business were also down by a double-digit percentage. »*

— **[Phrase_Precedente]** *« For instance, Henkel and Unilever witnessed a decline in shampoo consumption in the first quarter of 2020 when the coronavirus has just started to transmit worldwide. »*

— **[Titre_Depeche]** *« Shampoo Market to Hit USD 39.58 Billion by 2028 ; Custom Made Hair Cleanser to Augment Market Growth, Says Fortune Business Insights™ »*

Dans cet exemple, deux sociétés sont mentionnées dans la [Phrase_Precedente] : « Henkel » et « Unilever ».Le [Titre_Depeche] discute du marché du shampoing en général, ce qui pourrait être un indice de résolution de coréférence ; il ne s'agit plus d'une société spécifique mais d'un marché en général.

6 Conclusions

A partir de ce travail, notre objectif était double : [1] constituer un corpus spécifique aux dépêches d'actualités en anglais dans le domaine de l'économie et des finances afin de pouvoir annoter les coréférences des noms de société [2] fournir des recommandations pour le traitement de la coréférence en corpus qui pourront servir de base d'entraînement à des ou-

tils automatiques ou à des analyses linguistiques. Les résultats obtenus sont déjà concluants et cette approche est assez facilement généralisable mais nous envisageons de travailler également sur des marqueurs linguistiques pour décider si il y a continuité ou discontinuité dans la chaîne de référence. Egalement, étendre notre étude non seulement à la phrase précèdente et au titre, mais à l'ensemble des phrases de la dépêche d'actualité serait une analyse supplémentaire intéressante pour mesurer la coréférence en fonction de la proximité textuelle des maillons coréférents.

Références

Ariel, M. Referring and accessibility. In *Journal of linguistics*, Volume 24.1.

Boudreau, S. *Résolution d'anaphores et identification des chaînes de coréférence selon le type de texte*. Thèse de doctorat, Université de Montréal.

Dany, B., A. Jean-Yve, V. Jeanne, et L.-H. Anaïs. Redonner du sens à l'accord interannotateurs : vers une interprétation des mesures d'accord en termes de reproductibilité de l'annotation. In *HAL. Open Science*.

Delaborde, M. *Analyse en corpus de chaînes de coréférence : la coréférence non-stricte à l'épreuve de la linguistique outillée*. Thèse de doctorat, Université de Sorbonne Nouvelle - Paris 3.

Gobert, E. et B. Fabre. Détection de coréférences de bout en bout en français. In *Actes JEP TALN RECICL 2017*.

Landragin, F. et B. Oberle. Identification automatique de chaînes de coréférences : vers une analyse des erreurs pour mieux cibler l'apprentissage. In *Actes PFIA 2018*.

Recasens, M., M.-C. De Marneffe, et C. Potts. The life and death of discourse entities : Identifying singleton mentions. In *Proceedings of NAACL-HLT 2013*.

Schnedecker, C. De l'intérêt de la notion de chaîne de référence par rapport à celles d'anaphore et de coréférence. In *Les cahiers de praxématique*.

Summary

This article presents a feedback carried out in an industrial framework on coreference resolution. The experience first consists of observing a corpus from sentences of economic news in order to observe and interpret the connections between two coreferential chains that refer to the same entity. More precisely, we observed coreferences in a specific kind of text: news in English with economic and financial topics. We studied two specific cases of coreferences: first one is about periphrasis (*e.g.* between a proper noun as "Globus Medical, Inc." and a noun phrase as "The medical device company") ; the second is about anaphora (*e.g.* between a proper noun as "Salona Global Medical Device Corporation" and a pronoun as "it"). Our experience then consisted of evaluating whether it was possible to link a coreference in a reliable way to its antecedent from two logical relationships: [1] to its previous sentence [2] to the title of the news.

LoGE: Expansion Locale-Globale de document non supervisée avec un moteur de recherche Extensible

Oussama Ayoub*,**, Ludovic Li**, Christophe Rodrigues**, Nicolas Travers**

*Seville More Hélory, Paris, France
**Léonard De Vinci Pole Universitaire, Research Center, Paris La Défense, France
prénom.nom@devinci.fr

1 Contexte

Avec la croissance continue des données textuelles que les systèmes d'information modernes doivent gérer, des solutions de recherche d'information sont nécessaires pour trouver efficacement le meilleur ensemble de documents pour une demande donnée. Pour résoudre ce problème, nous proposons un moteur de recherche extensible qui vise à générer une expansion des documents en s'appuyant sur des méthodes récentes d'apprentissage profond et mis en œuvre sur Elasticsearch. Pour générer de nouveaux mots pour un document, un modèle de langage masqué d'apprentissage profond est utilisé pour inférer des mots apparentés. La démonstration montrera à la fois l'extensibilité de notre cadre de génération d'expansions, l'efficacité de l'évaluation et l'impact de diverses expansions sur la correspondance des requêtes.

La recherche d'information a évolué de manière continue pendant quatre décennies. La recherche ad hoc a fait l'objet d'un large débat sur le problème de vocabulaire lié à l'inadéquation entre les requêtes et les documents en raison d'aspects terminologiques et synonymiques.

La représentation vectorisée du texte est un moyen de réduire ce problème en complétant l'information sémantique par des réseaux neuronaux. L'approche la plus représentée est le *plongement de mots* Mikolov et al. (2013); Devlin et al. (2019) qui prend en compte le contexte des mots. Cependant, ces approches sont moins efficaces sur les textes longs et doivent être réentraînées/transférées. De plus, elles nécessitent de traiter la requête afin de la projeter dans le même modèle d'espace vectoriel, ce qui entraîne une augmentation du temps de traitement.

Notre processus d'expansion de documents repose à la fois sur une vue locale qui propose de nouveaux termes en fonction d'un document, et sur une vue globale qui filtre les termes en fonction de la représentation du corpus, comme l'illustre la figure 1. Une des particularités de notre approche est de proposer un processus modulaire où les modèles peuvent être échangés par d'autres approches similaires. Notre démonstration vise à montrer cette ouverture.

Pré-filtrage des documents. Le premier module sélectionne les termes du document en appliquant une étape de nettoyage standard et en conservant les termes les plus représentatifs. Cette dernière étape peut être basée sur l'IDF mais aussi sur un résumé extractif.

Extension des documents. Le second module est une *génération locale* qui s'appuie sur un modèle externe pré-entraîné. Son objectif est de proposer des termes basés sur le document d'entrée mais uniquement pour les termes pré-filtrés. Le contexte local aide à prédire les termes

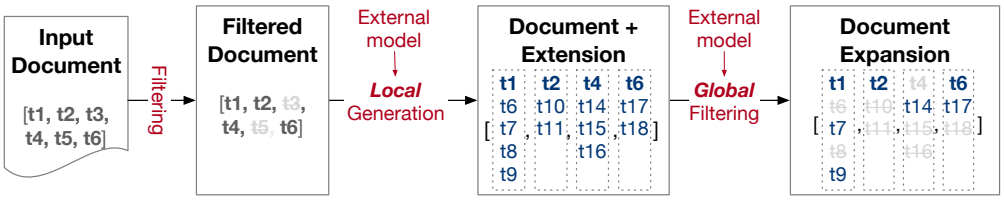


FIG. 1 – *LoGE - Processus d'expansion Locale-Globale*

pertinents sur la base d'un modèle BERT en fenêtrant le document d'entrée afin de gérer les longs documents (limitation de BERT).

Expansion des documents. Le dernier module filtre les termes générés en appliquant une étape de *Filtrage global* pour réduire le bruit et garantir une taille correcte du document. Il prend en compte le contexte général des documents afin de se concentrer sur un domaine donné. Nous utilisons un *plongement de mots* apporté par un modèle externe qui projette les mots dans un espace multidimensionnel (*par exemple,* ACP) et agit comme un filtre en décidant si un terme appartient à la dimension ou au cluster approprié.

Plus que modulaire, notre processus bénéficie d'une *approche non supervisée*. Les modèles externes ne sont pas nécessairement basés sur le corpus puisque leur but est de diversifier le document (*e.g.* ClinicalBERT Alsentzer et al. (2019), LegalBERT Chalkidis et al. (2020)). De plus, nous n'apprenons pas la correspondance entre les documents et les requêtes.

2 Le framework LoGE

Grâce au processus modulaire (indépendant) et aux expansions de documents (reposant sur des termes), nous pouvons connecter des moteurs de recherche standard comme `Elasticsearch` [1] pour faire évoluer notre système. Pour cela, nous avons conçu une architecture (Fig. 2) qui génère l'expansion des documents hors ligne avec des triggers tout en maintenant le système opérationnel.

Pile ELK. La pile ELK complète est utilisée pour importer des documents (`LogStash`) directement dans le cluster de nœuds `Elasticsearch` et produire un tableau de bord pour le suivi (`Kibana`). Les textes sont stockés sous forme de documents JSON qui seront étendus indépendamment par des triggers comme illustré ci-dessous.

```
{"id":1, "text":"t1 t2 t3 t4 t5 t6",
    "filter":{"f1":["t1","t2","t4","t6"]},
    "extension":{"f1_bertUncased":[[{"score":0.9, "term":"t1'"},
        {},{}], [{}], [{}], [{}]]},
    "expansion":{"f1_BU_closeAxis":[[{"score":0.9, "term":"t1'"}],
        [{}],[{}]]} }
```

API & UI. Une implémentation `python` fournit une API REST pour manipuler diverses extensions de documents avec des requêtes prédéfinies. Une interface utilisateur développée

1. `https///elastic.co`

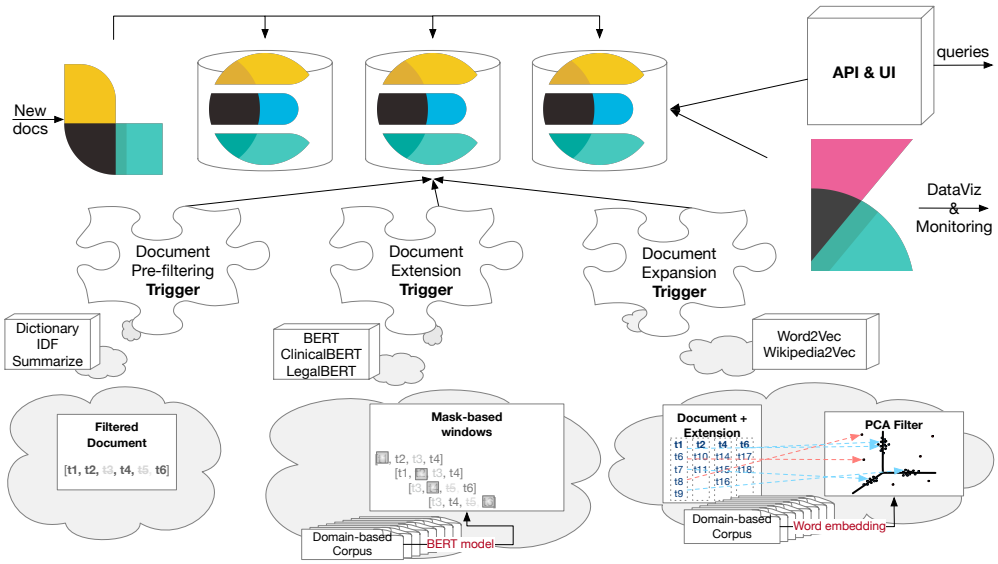


FIG. 2 – *L'architecture de* LoGE

en `react.js` permet de fournir différents types de requêtes. Elle met en évidence les mots correspondants afin d'améliorer le caractère explicatif de l'approche.

Trigger 1 Pré-filtrage des documents. Ce trigger surveille le contenu du corpus pour attraper tout document qui n'a pas encore été filtré. Chaque stratégie de filtrage génèrera une nouvelle clé dans "*filter*".

Trigger 2 Extension des documents. Ce trigger lourd utilise un modèle BERT [2] pour faire des prédictions. Pour que cela fonctionne correctement, nous devons masquer les termes du texte. Pour cela, une fenêtre glissante est appliquée sur le document pour garder le contexte local comme masque d'un terme (*i.e.,* termes filtrés du trigger 1). Les listes de prédictions sont stockées en tant que nouvelle clé dans ""*extension*". Les prédictions sont conservées pour le post-filtrage, les requêtes et l'explicabilité.

Trigger 3 Expansion des documents. Ce trigger vérifie le corpus pour voir si une extension a besoin d'être filtrée. Différentes stratégies peuvent être appliquées sur les extensions et stockées comme de nouvelles clés dans "*expansion*". Nous avons utilisé trois filtres différents basés sur une projection ACP (Wikipedia2Vec [3]) dont les termes retenus sont : 1) proches des axes ACP, 2) éloignés des axes, ou 3) proches du centroïde.

Toute l'architecture a été développée comme un cluster `Docker` avec différents services où nous pouvons régler le nombre de nœuds Elastic et de triggers (tout en gérant le parallélisme). Le code source est disponible en ligne [4].

2. `https://huggingface.co/google/bert_uncased_L-4_H-256_A-4`
3. `https://wikipedia2vec.github.io/wikipedia2vec/pretrained/`
4. `https://github.com/leonard-de-vinci/LoGE_DocExt_BERT-FILTER` (GNU GPL v2)

3 Démonstration de LoGE

Pour cette démonstration, nous nous appuyons sur deux ensembles de données spécialisées connus de la littérature : *Antique* Hashemi et al. (2020) et *NFCorpus* Boteva et al. (2016). Nous prévoyons de présenter 3 étapes principales :

1. La mise en place du framework LoGE en déployant le cluster et montrant l'aspect modulaire de notre approche. Nous portons une attention particulière au choix des stratégies d'expansion des documents, y compris en en mettant plusieurs en parallèle.
2. Une fois lancé, nous surveillerons la génération d'extensions de documents. Grâce à kibana, nous afficherons les filtres, les extensions et les expansions en temps réel. Plus intéressant encore, nous étudierons la répartition des termes entre les différentes stratégies pour révéler l'effet de BERT et de l'ACP.
3. Enfin, nous exécuterons des requêtes à partir des jeux de données correspondants. L'interface utilisateur illustrera l'impact de chaque étape de l'expansion du document en montrant les termes mis en évidence et le score BERT correspondant, ainsi que la nouvelle taille du document. Nous exécuterons les requêtes sur des documents filtrés, des documents étendus, des documents développés et n'importe quelle combinaison de ceux-ci ou même sans le document d'entrée lui-même. Il est intéressant de noter que BM25 a obtenu de meilleures performances avec notre expansion de documents que le TF-IDF standard.

Nous discuterons des résultats obtenus et donnerons notre avis sur notre nouvelle approche d'expansion de documents.

Références

Alsentzer, E., J. Murphy, W. Boag, W.-H. Weng, D. Jindi, T. Naumann, et M. McDermott (2019). Publicly available clinical BERT embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, Minneapolis, Minnesota, USA, pp. 72–78. Association for Computational Linguistics.

Boteva, V., D. Gholipour, A. Sokolov, et S. Riezler (2016). A full-text learning to rank dataset for medical information retrieval. In *Proceedings of the 38th European Conference on Information Retrieval (ECIR'16)*.

Chalkidis, I., M. Fergadiotis, P. Malakasiotis, N. Aletras, et I. Androutsopoulos (2020). LEGAL-BERT : The muppets straight out of law school. In *Findings of the Association for Computational Linguistics : EMNLP 2020*, pp. 2898–2904.

Devlin, J., M.-W. Chang, K. Lee, et K. Toutanova (2019). Bert : Pre-training of deep bidirectional transformers for language understanding.

Hashemi, H., M. Aliannejadi, H. Zamani, et W. B. Croft (2020). Antique : A non-factoid question answering benchmark. In J. M. Jose, E. Yilmaz, J. Magalhães, P. Castells, N. Ferro, M. J. Silva, et F. Martins (Eds.), *Advances in Information Retrieval*, Cham, pp. 166–173. Springer International Publishing.

Mikolov, T., K. Chen, G. S. Corrado, et J. Dean (2013). Efficient estimation of word representations in vector space. In *ICLR*.

Défi TextMine'23 - Reconnaissance d'entités d'intérêts dans les signatures d'e-mails

Kévin Cousot (Emvista), Cédric Lopez (Emvista), Pascal Cuxac (INIST-CNRS), Vincent Lemaire (Orange Labs)

1 Introduction

Le 21 octobre 2022, l'association Extraction et Gestion des Connaissances (EGC) a lancé le groupe de travail TextMine. Dans le cadre de ce groupe de travail, un objectif est de confronter l'état de l'art scientifique aux problèmes de text mining rencontrés par des industriels. Sous la forme de défis, le groupe de travail propose des jeux de données inédits et les partage avec la communauté scientifique. Le premier défi du groupe de travail TextMine a été lancé le 21 octobre en étroite collaboration avec la société Emvista, éditrice de logiciels fondés sur des technologies du Traitement Automatique du Langage Naturel, qui a fourni une partie des données. En particulier, la société s'intéresse à la structuration des informations véhiculées dans les e-mails.

Le défi proposé porte sur la reconnaissance d'entités d'intérêts dans les signatures d'e-mails dans le but de structurer l'information et de la stocker en base de données (par exemple un système de gestion de la relation client). Il s'agira de s'approcher de conditions réelles dans lesquelles les données disponibles à l'entraînement ne reflètent pas nécessairement la distribution des données auxquelles le système devra faire face en production. Le participant qui obtiendra le meilleur résultat se verra remettre un prix de 300 euros.

La suite du document s'organise comme suit. Nous commencerons par décrire tâche (section 2) et les données à utiliser (section 3). Les modalités d'évaluation et de participation sont ensuite expliquées section 4.

2 Tâche

La tâche se formule comme un problème de classification de tokens à 13 classes. On prend en entrée un texte brut dans lequel les entités d'intérêts sont identifiées. Il s'agit alors d'attribuer une étiquette à chacune des entités.

Donnée d'entrée :	**Résultat attendu :**
Anna$_{?}$ Dupont$_{?}$	Anna$_{\text{Human}}$ Dupont$_{\text{Human}}$
Directeur$_{?}$ Général$_{?}$	Directeur$_{\text{Function}}$ Général$_{\text{Function}}$
Téléphone : 01.55.52.12.96$_{?}$	Téléphone : 01.55.52.12.96$_{\text{Phone_Number}}$
www.anywaythewall.com$_{?}$	www.anywaythewall.com$_{\text{Url}}$

3 Les données

Face à l'absence de données annotées, le groupe de travail a produit trois jeux de données à l'aide de différentes stratégies. Toutes ces données sont partagées avec la communauté scientifique.

- Un Jeu de données authentique (JDA dans la suite) : un jeu de données composé de signatures authentiques pseudonymisées ;

Classe	Définition
Human	Noms et prénoms des personnes qui figurent dans la signature. Ex : Martin Dupond
Organization	L'organisation à laquelle l'auteur de la signature est rattachée.
Function	L'ensemble des fonctions assignées à la personne identifiée dans la signature. Ex : conseiller, assistant, professeur.
Project	Projet dans lequel la personne est impliquée. Ex : Comm & Partenariats, Master 2 Informatique, Direction de l'innovation.
Location	Bâtiments, bureaux, villes, numéros et noms des rues. Ex : 24 avenue Jean Jaurès, Montpellier
Reference_CEDEX	Courrier d'entreprise à distribution exceptionnelle. Ex : Cedex 05
Reference_CS	Course spéciale. Ex : CS 39521.
Reference_Code_Postal	Code postal. Ex : 34080.
Phone_Number	Numéro de télphone ou de fax. Ex : 01.23.45.67.89
Email	Adresses e-mails. Ex : martin.dupond@bboite.com
Url	URL, typiquement vers le site web de l'entreprise ou la personne. Ex : www.anywaythewall.com
Social_Network	Nom des réseaux sociaux. Ex : LinkedIn, Facebook, Twitter...
Reference_User	Identifiant d'une personne ou d'une organisation sur un réseau social. Ex : @Kalypso_immobilier

Table 1: Classes annotées dans les jeux de données.

- Jeu de données réaliste (JDR dans la suite) : un jeu de données composé de signatures construites manuellement par la société Isahit, plateforme de labellisation éthique des données pour l'IA[1] ; ce jeu de données contient des signatures réalistes, c'est-à-dire proches des signatures authentiques observées, mais non authentiques (elles n'ont jamais été utilisées dans des échanges d'e-mails) ;
- Jeu de données factice (JDF dans la suite) : un jeu de données composé de signatures créées automatiquement à partir d'une API de génération de fausses identités[2]

Les entités présentes dans les signatures sont annotées à l'aide d'une typologie de 13 classes, présentées dans le tableau 1.

3.1 Jeu de données authentique (JDA)

Face à l'absence de signatures authentiques parmi les jeux de données disponibles, la société Emvista a mis en place un formulaire Web permettant le "don" de signatures pendant 6 mois (Bendahman et al., 2022).

Pour préserver l'identité des contributeurs, un processus de pseudonymisation a été appliqué. Chaque entité d'intérêt a été remplacée par une autre entité de même classe afin d'assurer la cohérence du jeu de données. Certaines classes ont été traitées manuellement, d'autres de façon automatique. Notons également que les fonctions des personnes n'ont pas été remplacées, sauf dans le cas où la description permettait d'identifier la personne.

Un exemple de pseudonymisation est donné ci-dessous :

[1] https://fr.isahit.com/
[2] https://www.fakenamegenerator.com/

Classe	Quantité d'annotations
Human	1196
Organization	1537
Location	2680
Phone_Number	688
Function	1449
Email	344
Url	303
Social_Network	28
Reference_User	11
Reference_Code_Postal	349
Project	124
Reference_CEDEX	146
Reference_CS	33

Table 2: Nombre d'annotations dans JDA.

Signature:

Mary Margho
Directeur Général
Téléphone : +33.(0)1.52.62.32.65
6 Rue Jean-Paul Montagne
75001 Paris
www.saveprogramma.com

Signature pseudonymisée :

Anna Dupont
Directeur Général
Téléphone : 01.55.52.12.96
72, Rue Paul-Marie L'abbé
34090 MONTPELLIER
www.anywaythewall.com

Au total, le JDA contient 606 signatures. La répartition des classes utilisées est indiquée dans le tableau 2.

3.2 Jeu de données réaliste (JDR)

Le jeu de données réaliste a été produit par la société Isahit avec un suivi régulier par les organisateurs du défi. Le cahier des charges fourni initialement à Isahit contenait des contraintes émises à partir de l'observation des signatures authentiques (cf. section 3.1).

D'une part, il a été demandé que la taille des signatures générées soient comprise entre 2 et 132 tokens avec une moyenne d'environ 46 tokens sur la totalité du jeu de données produit. Il a également été demandé qu'entre 1 et 3 tokens par signature ne soit pas catégorisable dans l'une des classes prédéfinies (par exemple « Tel : » ou encore la coordination « et » entre deux noms de personnes). Enfin, des contraintes d'ordre statistiques ont été émises afin d'approcher les distributions observées dans le JDA.

Le jeu de données a fait l'objet d'une vérification de la qualité des annotations :

1. pour chaque signature, vérification automatique du respect des contraintes imposées sur l'utilisation des classes. Ce point a déclenché des aller-retours avec la société afin d'obtenir un respect total des contraintes énoncés dans le cahier des charges;

2. un échantillon de 50 signatures prises aléatoirement a été évalué manuellement : 100% des annotations observées sont correctes.

Au total, le JDR contient 473 signatures. La répartition des classes utilisées est indiquée dans le tableau 3.

Classe	Quantité d'annotations
Human	971
Organization	1023
Location	2533
Phone_Number	473
Function	567
Email	297
Url	227
Social_Network	18
Reference_User	9
Reference_Code_Postal	269
Project	98
Reference_CEDEX	150
Reference_CS	56

Table 3: Nombre d'annotations dans JDR.

Classe	Quantité d'annotations
Human	1023
Organization	943
Location	2150
Phone_Number	371
Function	0
Email	371
Url	0
Social_Network	0
Reference_User	0
Reference_Code_Postal	367
Project	0
Reference_CEDEX	0
Reference_CS	0

Table 4: Nombre d'annotations dans JDF.

3.3 Jeu de données factice (JDF)

Le jeu de données factices (JDF) est composé de signatures créées automatiquement. La génération s'appuie sur des patrons définis manuellement, une API de création de fausses identités[3] et des heuristiques introduisant de la variabilité au sein des entités. L'API ne fournissant pas certaines classes, celles-ci sont absentes du JDF : Project, Url, Reference_User, Reference_CEDEX, Reference_CS et Function.

Au total, 500 signatures ont été générées.

3.4 Format

Les données sont distribuées en format JSON. Chaque signature possède trois champs :

- "identifier": l'identifiant, numérique, de la signature
- "text": le texte brut de la signature
- "annotations" : la liste des annotations d'entités d'intérêts portant sur la signature

Une annotations contient quatre champs :

[3] https://www.fakenamegenerator.com/

- "form" : la forme du token annoté
- "label" : la classe associée au token
- "begin" : l'index du début du token (inclus)
- "end" : l'index de fin du token (exclu)

Exemple :

```
[
  {
    "identifier" : 0,
    "text" : "Faustin Dupont",
    "annotations" : [
      {
        "form" : "Faustin",
        "label" : "Human",
        "begin" : 0,
        "end" : 7
      },
      {
        "form" : "Dupont",
        "label" : "Human",
        "begin" : 8,
        "end" : 14
      }
    ]
  }
]
```

4 Évaluation

Le défi consiste à obtenir la meilleure F-mesure[4] sur la tâche de reconnaissance d'entités d'intérêts dans le jeu de test. Tout système est le bienvenu : symbolique, connexionniste, à base de connaissances ou d'apprentissage etc.

Pour l'entraînement, les participants ont accès au JDF ainsi qu'au JDR. La répartition des classes est donnée tableau 5. Les participants ont la liberté d'utiliser ces jeux de données comme ils le souhaitent mais sans y apporter de modification. Les données sont téléchargeables sur le Github suivant : https://github.com/Emvista/Challenge_TextMine_2023.

Le jeu de données utilisé pour l'évaluation finale est le JDA. Celui-ci sera ajouté au dépôt github le 8 janvier 2023. Dès lors, les participants ont jusqu'à la clôture du défi, le 10 janvier 2023, pour soumettre leurs résultats par mail à l'adresse textmine@emvista.com.

Les résultats d'évaluation leur seront communiqués après chaque soumission, mais le classement des participants ne sera dévoilé qu'à la remise du prix, le jour de l'atelier TextMine au sein de la conférence EGC, le 17 janvier 2023.

Le jeu de test respecte le même format que les jeux d'entraînement à l'exception du fait que le label y est absent. Ce champ est à remplir par les participants et doit être présent dans les résultats soumis à évaluation (voir listing 1 et 2).

[4] 2*(precision*rappel) / (précision + rappel)

Classe	Quantité d'annotations
Human	1994
Organization	1966
Location	4683
Phone_Number	844
Function	567
Email	668
Url	227
Social_Network	18
Reference_User	9
Reference_Code_Postal	636
Project	98
Reference_CEDEX	150
Reference_CS	56

Table 5: Nombre d'annotations dans les données d'entraînement JDR+JDF (11916 annotation).

Listing 1: Données de test

```
[
  {
    "identifier" : 0,
    "text" : "Faustin Dupont",
    "annotations" : [
      {
        "form" : "Faustin",

        "begin" : 0,
        "end" : 7
      },
      {
        "form" : "Dupont",

        "begin" : 8,
        "end" : 14
      }
    ]
  }
]
```

Listing 2: Soumission du participant

```
[
  {
    "identifier" : 0,
    "text" : "Faustin Dupont",
    "annotations" : [
      {
        "form" : "Faustin",
        "label" : "Human",
        "begin" : 0,
        "end" : 7
      },
      {
        "form" : "Dupont",
        "label" : "Human",
        "begin" : 8,
        "end" : 14
      }
    ]
  }
]
```

References

Nihed Bendahman, Kevin Cousot, and Cédric Lopez. Reconnaissance d'entités d'intérêt dans les signatures d'e-mails à partir d'un jeu de données authentique. *TextMine'22*, 2022.

Participation d'EDF R&D au Défi Textmine 2023 : Reconnaissance d'entités d'intérêts dans les signatures d'e-mails

Philippe Suignard*, Leila Hassani*, Meryl Bothua*

*EDF R&D
7, boulevard Gaspard Monge 91120 Palaiseau
prenom.nom@edf.fr

Résumé. Cet article présente la contribution d'EDF R&D au défi Textmine 2023 qui a pour objectif la « Reconnaissance d'entités d'intérêts dans les signatures d'e-mails ». Deux méthodes ont été mises en œuvre : l'une s'appuyant sur une extraction de features et l'entraînement d'un classifieur, l'autre en s'appuyant sur le framework Flair. L'article présente les résultats obtenus sur les données de test.

1 Introduction

Le défi Textmine 2023 (Cousot et al., 2023) a pour objectif la **« Reconnaissance d'entités d'intérêts dans les signatures d'e-mails »**. Pour mener à bien ce défi, les organisateurs ont récolté un jeu de données dans le cadre d'un crowd-sourcing. Ce jeu de données (JDA) a été anonymisé ; il sert à l'évaluation finale des résultats. A partir du JDA, un jeu de données réaliste a été construit manuellement (JDR). Pour augmenter la taille des données d'apprentissage, les organisateurs ont également produit un jeu de données factice (JDF). Les deux jeux de données JDR et JDF peuvent être utilisés pour entraîner un système qui sera donc évalué sur le JDA. Les données sont constituées de 13 étiquettes ou classes : email, function, human, location, organization, phone number, project, reference cedex, reference CS, reference code postal, URL, reference user, social network.

La suite du document présente les méthodes développées par EDF R&D dans le cadre de ce défi (partie 2 et partie 3). La partie 4 présente les résultats obtenus.

2 Extraction de features et entraînement d'un classifieur

L'approche retenue ici est une approche de type classification : les données étant prédécoupées en « token », la méthode consiste à attribuer une des 13 étiquettes possibles à chaque token. De manière générale, les signatures de mails ne sont pas du tout structurées, mais elles comprennent bien sûr des éléments incontournables comme le nom de la personne, son numéro de téléphone, son adresse, etc. Deux types d'information varient dans les signatures :

— L'ordre de ces informations, même si le nom de la personne a, en général, tendance à être présent en premier ;

— La manière de coder les mots ou tokens eux-mêmes comme par exemple les numéros de téléphone que l'on peut trouver sous les formes suivantes : 03.18.38.37.37 (avec des points), + 03 81 20 48 27 (des blancs), + 33 01 77 83 74 05 (un indicatif), +33 0365962110 (des chiffres collés), 03 46 69 07 08 (séparés), etc.

L'approche proposée pour étiqueter les tokens se situe dans la lignée des travaux de (Bendahman et al.), les organisateurs du défi et sera découpée en deux parties :

— La partie extraction de « features » ou descripteurs : cette partie consiste à extraire des caractéristiques à partir des mots eux-mêmes et de leur position dans la signature ;
— La partie apprentissage qui consiste à entraîner un classifieur à partir des « features » précédentes.

2.1 Extraction de « features »

Les signatures d'e-mails sont découpées en lignes ou chunks. Chaque chunk est ensuite découpé en tokens. Dans l'exemple suivant, la signature est découpée en 8 lignes ou chunks :

1. Faustin Chabot
2. Adresse : 19 rue Descartes 94370 Sucy-en-Brie (France)
3. Cedex 9 CS 12468
4. Data Engineer / Algorithm XZ Project
5. faustinchabot@teleworm.com / Tel : +33 0134354919
6. Linkedin : https ://fr.linkedin.com/in/fauchab
7. Teleworm France
8. teleworm.france.com

Quelques remarques :

— Les étiquettes sont attribuées aux tokens et non aux chunks : Faustin et Chabot sont étiquetés séparément (ici « Human ») ;
— Les chunks peuvent contenir plusieurs étiquettes différentes : le deuxième chunk comprend 2 étiquettes différentes (« Location » et « Reference Code Postal ») ;
— Les mots d'un chunk ne sont pas tous étiquetés. Dans le 2ème chunk, le mot « Adresse » n'a pas d'étiquette.

2.1.1 Features pour chaque Token

Pour chaque token à classer, la méthode commence par calculer les 20 features suivantes :

— **Longueur du token** : nombre de lettres qui constituent le token. Cette feature sera importante pour discriminer des tokens très longs comme les URL ;
— **Cedex** : vaut 1 si le token est égal au mot « cedex ». Dans le cas de « Cedex 09 », les mots « Cedex » et « 09 » sont étiquetés « Reference CEDEX » chacun. Cela permet ainsi d'étiqueter directement le mot « Cedex » et de désambiguïser « 09 », Cf partie suivante ;
— **Nombre de blancs** : nombres de blancs contenus dans le token ;
— **Nombre de points** : nombres de points dans le token. (Ces 2 features permettent de discriminer les numéros de téléphones et les URL, par exemple) ;

— **Arobase** : vaut 1 si le token contient le caractère arobase, notamment pour discriminer les adresses mail ;
— **Démarre par une majuscule** : 1 ou 0 selon que le mot démarre par une majuscule ou non ;
— **Tout en majuscule** : 1 si tous les caractères du token sont en majuscule, 0 sinon.
— **RS** pour réseau social : 1 si le token est égal à un Réseau Social comme Facebook, Instagram ou LinkedIn et 0 sinon ;
— **Numérique** : 1 si tous les caractères qui composent le token sont des chiffres ;
— **Localisation** : 1 si le token fait partie d'une liste de mots de localisation comme : « rue », « avenue », « boulevard », etc. ;
— **Fonction** : 1 si le token fait partie d'une liste de mots associés à des fonctions comme responsable, manager, gestionnaire, assistant(e), chef(fe), etc. et 0 sinon ;
— **Organisation** : 1 si le token fait partie d'une liste de mots associés à des mots d'organisation comme « cabinet », « clinique », « hôpital », etc. ;
— **Projet** : 1 si le token vaut « project » (seul mot discriminant) ;
— **Numéro de chunk** : numéro du chunk (ou ligne) dans lequel est situé le token. Généralement, la signature commence par le nom de la personne (premier chunk), puis l'adresse ou le projet, etc. Ce numéro de chunk ou ligne permet de désambiguïser les classes ;
— **Longueur du chunk** : nombre de caractères du chunk dans lequel se trouve le token ;
— **Nb mot chunk** : nombre de mots (séparateur espace) du chunk dans lequel se trouve le token ;
— **Pos in chunk** : position du token dans le chunk (1 pour le premier mots, etc.) ;
— **Mail** : 1 si le mot finit par « .com » ou « .fr », 0 sinon ;
— **Tiret** : 1 si le token contient « - » et 0 sinon (permet de cibler les noms de communes contenant des tirets comme « Sucy-en-Brie », par exemple) ;
— **Adresse** : 1 si le chunk dans lequel se trouve le token est du type : 2 xxx 75100 YYY.

Une 21^{eme} feature, **nb chunk**, est égale au nombre de chunks de la signature. Cette valeur est la même pour tous les tokens de la signature.

2.1.2 Features de position

Une des limitations des « features » définies précédemment, est leur côté non contextuel. Par exemple, le token « de » devra avoir pour étiquette « Location » dans le cas « rue **de** la Paix », mais « Function » dans le cas « chef **de** projet ». Même chose pour un nombre comme 2 : « Location » dans « **2** rue de la Paix », mais « Reference Cedex » dans « Cedex **2** ». Pour résoudre ce problème, on va tripler le nombre de features par token, en affectant à chaque token :

— les 20 features du token en question ;
— les 20 features du token précédent ;
— les 20 features du mot suivant.

Ainsi les features des tokens précédant et suivant le token courant vont contribuer à mieux choisir la bonne étiquette à attribuer au token courant [1].

1. Le premier token d'un chunk n'a pas de token précédent. Le dernier token d'un chunk n'a pas de token suivant. Dans ces cas de figure, des tokens vides sont ajoutés (20 valeurs "-1").

2.2 Entraînement d'un classifieur

Les 61 features précédentes sont ensuite utilisées pour entraîner un classifieur à l'aide du logiciel Weka (Hall et al., 2009). Plusieurs classifieurs sont testés. "Random Forest" est celui qui obtient les meilleurs résultats. Deux entrainements différents sont réalisés par validation croisée sur 5 plis, l'un avec uniquement les données du JDR et l'autre avec les données du JDR et du JDF. Les résultats obtenus dans la phase d'apprentissage sont les suivants :

	JDR	JDR+JDF
Accuracy	96,1	97,6
Précision	96,1	97,5
Rappel	96,1	97,6
F-mesure	95,8	97,4
Individus	6691	11916

TAB. 1 – *Résultats obtenus sur les données d'apprentissage en %.*

3 Détection des entités avec Flair

3.1 Formatage des données

L'entraînement d'un modèle avec FLAIR (Akbik et al., 2018) et (Akbik et al., 2019) requiert un format particulier qui permet d'associer chaque *token* à son annotation séparée par une tabulation. Par défaut, un *token* sans annotation correspondante est associé à une valeur de O. Tous les textes annotés sont alors regroupés dans un même fichier texte en fonction du jeu de données (*train*, *test* ou *dev*) auquel il appartient.

Afin d'obtenir un corpus plus conséquent, nous avons fait le choix de mélanger les jeux de données JDR et JDF avant de diviser le corpus en *train* (70%), *test* (10%) et *dev* (20%). Les données originales étant en format JSON, le formatage a été effectué en Python grâce au module *json*.

Lors de l'importation des fichiers contenant les données, le module FLAIR considère tout espace ou tabulation comme marquant la présence d'une autre colonne. Cela signifie qu'un numéro de téléphone annoté sous la forme `+33 01020304 Phone_Number` sera traité comme trois colonnes différentes. Puisque nous récupérons la deuxième colonne afin d'établir les catégories uniques à prédire, en l'occurence, `01020304` sera considéré comme une catégorie à part entière, à l'instar des catégories `Human` ou `Reference_CEDEX`. Afin de parer ce problème, nous devons donc passer par une étape de normalisation des numéros de téléphone. Nous avons fait le choix de ne garder ni les espaces, ni les points, ni les tirets entre les numéros. Nous avons également retiré les parenthèses autour des indicatifs lorsqu'il y en avait.

La tokenisation est ensuite faite à partir des espaces, sauf dans le cas de certains élements des catégories `Project` et `Organization`. En effet, dans le fichier JSON, ces éléments sont décomposés et chaque token est alors identifié au même label. Par exemple, dans le jeu de données JDF, l'organisation "*d.e.m.o*" est tokenisée de la manière suivante : *'d', '.', 'e', '.', 'm',*

'.', 'o', où chacun des *tokens* est associé à la catégorie `Organization`. Nous avons donc pris en compte cette annotation lors de la tokenisation.

3.2 Entraînement du modèle

L'entraînement a été réalisé grâce au module FLAIR, un framework open-source basé sur la librairie PyTorch. FLAIR permet la création d'un objet Corpus à partir de nos trois documents texte *test*, *train*, *dev*. Nos documents sont divisés en deux colonnes : la colonne texte, et la colonne NER.

La vectorisation des textes est faite grâce à des modèles de plongement de mots pré-entraînés. L'une des innovation de FLAIR est que le module permet d'empiler plusieurs types de plongements. Nous avons fait le choix d'utiliser à la fois des *embeddings* positionnels (WordEmbeddings + CharacterEmbeddings) et des *embeddings* contextuels (FlairEmbeddings). Ces derniers sont entraînés à partir d'un LSTM bi-directionnel qui permet la vectorisation d'un mot en fonction de son contexte à droite et à gauche.

Nous avons fait le choix d'initialiser la taille de la couche cachée du bi-LSTM à 200, ainsi que d'utiliser la couche CRF (Conditional Random Field) en sortie. Cette couche introduit une étape de modélisation statistique basée sur le concept de Champ aléatoire de Markov. Elle est parfaitement adaptée à la prise en compte du contexte lors des prédictions.

3.3 Evaluation et Résultats

Avec un taux d'apprentissage de 0,1, une *batch size* de 64 et 50 epochs effectuées sur les données d'entraînement, notre meilleur modèle nous permet d'obtenir une F-mesure de 94,5% sur les données de validation et une F-mesure de 89,5% sur les données de test qui comprend 10% des données de JDR et JDF.

	dev	test (JDR+JDF)
précision	98,6	89,2
rappel	90	89,8
F-mesure	98,6	89,5

TAB. 2 – *Tableau des métriques pour les données de dev et de test JDR+JDF en %.*

Les très bon résultats sur le corpus de *dev* par comparaison avec le corpus de *test* s'expliquent par le formatage nécessaire à l'entraînement avec FLAIR développé à la Section 3.1. En effet, la catégorie `O` qui permet d'annoter les mots qui n'ont pas d'entité associé est considérée lors du calcul des métriques de précision et rappel sur le corpus de *dev*, mais pas celui de *test*. La catégorie `O` atteint un score de 100% de précision et 95% de rappel. Ceci s'ajoute à la précision de 100% de `Project` sur les données de *dev* (contre 0% pour les données de *test*). Ces deux éléments compensent les mauvais résultats sur le rappel de `Project` du corpus de *dev* (0%).

Afin de déterminer les erreurs les plus fréquemment commises par le modèle, nous pouvons

également observer les métriques obtenues pour chaque entité. Le tableau ci-dessous récapitule la précision et le rappel obtenus par entité.

	Précision	Rappel
Email	100	98,4
Function	91	93,7
Human	99,5	100
Location	98,7	99,8
Organization	97,3	95,3
Phone_Number	100	100
Project	0	0
Reference_CEDEX	100	100
Reference_CS	100	100
Reference_Code_Postal	100	100
Url	94,4	100

TAB. 3 – *Précision et rappel pour chaque entité du jeu de test JDF+JDR en %.*

Immédiatement, nous pouvons identifier l'entité `Project` comme la catégorie la moins reconnue par le modèle. En effet, c'est une entité peu régulière et facilement confondue avec l'entité `Function`, ce qui rend l'identification de *features* nécessaires à l'extraction difficile.

Nous remarquons également l'absence des entités `Social_Network` et `Reference_User` dans le tableau des résultats. Ceci s'explique par le nombre d'occurrence faible de ces entités dans les corpus originaux (seulement 9 occurrences de `Reference_User` et 18 occurrences de `Social_Network` dans le corpus JDR, et aucune dans le corpus JDF). Puisque la division des corpus en *test*, *train* et *dev* a été faite de manière aléatoire, ces entités n'ont pas été incluses dans le corpus de *test*.

Nous pouvons également regarder de plus près les annotations fournies par le modèle afin d'identifier les éléments qui ont posé problème. Prenons cet exemple du corpus de *test* :

Gaetane Dupuis
The Flying Hippo
54 Rue St Ferréol 57070 Metz

Les annotations du modèle sont les suivantes :

Gaetane [Human] Dupuis [Human]
The [Organization] Flying [Organization] Tiger [Organization]
54 [Location] Rue [Location] St [Location] Ferréol [Location] 57070 [Reference_Code_Postal] Metz [Location]

Les entités `Human` et `Location` et `Reference_Code_Postal` sont en général bien reconnues par le modèle. C'est aussi le cas des numéros de téléphone – à condition de passer d'abord par un pré-processing similaire à celui effectué sur les corpus de test, de validation et d'entraînement (Section 2.1). Les emails sont également assez bien reconnues, bien que les

résultats empiriques montrent que le modèle a tendance à surprédire les emails aux dépens des url. La plus grosse faiblesse du modèle est son inabilité à reconnaître les *tokens* qui ne sont en fait pas des entités, et donc annotés `O` dans nos corpus. Prenons cet exemple :

Amélie [Human] Pinneau [Human]
40 [Location] rue [Location] Marguerite [Location] 94300 [Reference_Code_Postal] Vincennes [Location]
Tél [Reference_CEDEX] : [Reference_CS] 01 27 34 62 01 [Phone_Number]
Email [Reference_User] : [O] ameliepinneau@gmail.com [Email]
Assistante [Function] sociale [Function]
ONG [Function] Fall [Project] In [Project] Love [Project] **- [Project]** filong.fr [Url]

Les erreurs du modèle sont en gras.

Les erreurs principales se produisent sur les éléments qui ne font pas parti des entités à reconnaître, en l'occurence *Tél*, *:*, *Email* et -. Pourtant, le modèle est bien capable d'annoter ces éléments, puisqu'il catégorise correctement l'élément *:* qui suit *Email*.

Une autre erreur fréquente observée est la confusion entre `Function` et `Project`. Cette confusion se produit également entre les entités `Url` et `Reference_User` ainsi qu'`Email` et `Url`. Plus surprenant et difficile à expliquer, il arrive que le modèle identifie les éléments `Reference_User` en tant qu'`Human`.

4 Résultats

Pour rappel :
— Run 1 correspond à la méthode 1 entraînée sur le JDR ;
— Run 2 correspond à la méthode 1 entraînée sur le JDR et le JDF ;
— Run 3 correspond à la méthode 2 (Flair).
Les résultats obtenus sont présentés dans le tableau suivant :

	Run1	Run2	Run3
Accuracy	68,17	68,32	49,64
Précision	69,50	69,78	72,11
Rappel	68,17	68,32	49,64
F-mesure	67,21	67,24	42,77

TAB. 4 – *Résultats obtenus sur les données de test (JDA) en %.*

On constate d'abord que les scores obtenus sur le test sont très inférieurs aux scores obtenus sur le corpus d'apprentissage. Pour ce qui est de la méthode 1, cela peut s'expliquer en partie par le formatage des signatures qui sont très bien découpées dans les corpus d'apprentissage mais qui sont moins "propres" dans le corpus de test. Dans le JDA, il a beaucoup de "\n" superflus. De même, un certain nombre d'adresses sont coupées par un "\n" (avec le numéro et le nom de la rue d'un côté mais le code postal et la ville de l'autre). Un prétraitement a

été réalisé pour corriger ou amoindrir ces problèmes, mais cela n'a pas suffit. A contrario, environ 10% des signatures (52/606) sont codées sur une seule ligne, le séparateur de nature de données devenant le tiret. Dans d'autres signatures, certaines séparations sont assurées par "\n" et d'autres par un tiret comme "Dominique Faubert - Directeur d'exploitation - Responsable de site \n DominiqueFaubert@armyspy.com". Pour améliorer les résultats, il faudrait améliorer le découpage des signatures.

Les très faibles écarts obtenus entre les run 1 et run 2 montrent que l'apport du corpus factice (JDF) est assez minime. Les signatures qu'il propose semble trop "régulières".

Quand on analyse les score obtenus catégorie par catégorie, on se rend compte que les scores sont très corrects sur certaines comme email, Human, Location, url ou phone_number, par contre elles chutent sur function, organization ou project.

Pour la partie Flair, nous soupçonnons que les mauvais résultats sont en partie dus à la manière dont le test a été effectué — token par token, par opposition au test effectué sur les données JDR+JDF, qui a été effectué sur des phrases entières. En effet, le modèle entraîné se reposant sur le contexte, l'absence de contexte autour d'un token pourrait expliquer certaines confusions du modèle, notamment les très mauvaises précisions d'Human, de Reference_Cedex, d'Url et d'Email, ainsi que les mauvais rappels pour les entités Reference_CS et Location. Cette théorie semble vérifiée par les tests effectués en aval sur les phrases entières, avec leur contexte, qui ont des résultats beaucoup plus pertinents que ceux obtenus lors du test effectué sur chaque token séparément.

5 Conclusion

L'équipe texte de la R&D d'EDF a participé à ce nouveau défi de l'atelier TextMine dans le cadre de la conférence EGC (Extraction et Gestion des Connaissances). Cette campagne nous a permis de tester des méthodes de détection d'entités d'intérêts dans les signatures d'e-mails. Ce sujet est très important pour EDF. Nous travaillons à la détection de données à caractère personnel afin d'en assurer la pseudonymisation et ainsi respecter le RGPD. Participer à ce défi est pour nous l'occasion d'échanger sur des méthodes de traitement automatique du langage avec des universitaires et des industriels.

Références

Akbik, A., T. Bergmann, D. Blythe, K. Rasul, S. Schweter, et R. Vollgraf (2019). Flair : An easy-to-use framework for state-of-the-art nlp. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics (demonstrations)*, pp. 54–59.

Akbik, A., D. Blythe, et R. Vollgraf (2018). Contextual string embeddings for sequence labeling. In *Proceedings of the 27th international conference on computational linguistics*, pp. 1638–1649.

Bendahman, N., K. Cousot, et C. Lopez. Reconnaissance d'entités d'intérêt dans les signatures d'e-mails à partir d'un jeu de données authentique. *TextMine'22*.

Cousot, K., C. Lopez, et V. Lemaire (2023). Défi textmine'23 - reconnaissance d'entités d'intérêts dans les signatures d'e-mails. actes de l'atelier textmine'23, p. à paraître, conférence extraction et gestion des connaissances 2023 (egc'23), lyon.

Hall, M., E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, et I. H. Witten (2009). The weka data mining software : an update. *ACM SIGKDD explorations newsletter 11*(1), 10–18.

Summary

This article presents the contribution of EDF R&D to the Textmine 2023 challenge which aims at "Recognition of entities of interest in e-mail signatures". Two methods have been implemented: one based on feature extraction and training of a classifier, the other based on the Flair framework. The article presents the results obtained on the test data.

GREYC@TextMine2023 : Reconnaissance d'entités nommées dans les signatures d'e-mails

Tanguy Gernot, Emmanuel Giguet

Normandie Univ, UNICAEN, ENSICAEN, CNRS, GREYC
14000 Caen, France
{prenom.nom}@unicaen.fr

Résumé. Cet article présente notre contribution au défi TextMine'23 portant sur la "Reconnaissance d'entités d'intérêts dans les signatures d'e-mails". La performance de notre système atteint 99% et 100% en f-mesure sur les jeux de données de mise au point et 83% de f-mesure sur le jeu de validation.

1 Introduction

Le défi TextMine'23 (Cousot et al., 2023) "Reconnaissance d'entités d'intérêts dans les signatures d'e-mails" est une initiative du groupe de travail TextMine dont l'objectif est de confronter l'état de l'art scientifique aux problèmes d'analyse de données textuelles rencontrés par les industriels. TextMine'23 est centré sur la reconnaissance d'entités d'intérêts dans les signatures d'e-mails dans le but de structurer l'information et de la stocker en base de données.

Suivant la tradition du traitement automatique des langues et des documents de notre équipe (Giguet et Lucas, 2022 ; Giguet et Lejeune, 2021 ; Giguet et Lucas, 2010), nous avons choisi de réaliser une chaîne de traitement complète, depuis la segmentation des signatures jusqu'à la production du fichier d'annotations au format attendu. Notre participation nous a permis d'explorer la dimension multilingue de l'analyse de signature, la sélection des unités d'analyse pertinentes, ainsi que le calcul original de chaîne de coréférence. La performance de notre système atteint 99% et 100% en f-mesure sur les jeux de données de mise au point et 83% de f-mesure sur le jeu de validation.

2 Travaux précédents

L'analyse des e-mails est sujet de recherches depuis le milieu des années 90. La question de la classification automatique a notamment fait l'objet de nombreux travaux, que ce soit pour la détection de courriers indésirables, le routage automatique des courriers entrants vers le bon interlocuteur, ou le classement thématique dans une arborescence. De nombreuses synthèses sont disponibles et la question de la segmentation des e-mails est une question régulièrement soulevée.

Concernant l'identification de la signature des e-mails et de leur analyse en constituants, (Bendahman et al., 2022) propose un état de l'art mettant en avant les travaux de (Chen et al., 1999 ; Carvalho et Cohen, 2004 ; Estival, 2008 ; Li et al., 2015). L'identification de la signature y est réalisée soit dans une perspective d'extraction d'informations ciblées, soit dans une perspective d'élimination de son contenu pour ne pas perturber l'analyse du corps de l'e-mail.

Après avoir constitué de manière participative un corpus de signatures d'e-mails et effectué leur pseudonymisation, (Bendahman et al., 2022) effectuent un comparatif d'algorithme d'apprentissage automatique (SVM, CRF, Bi-LSTM, Bert) pour la classification des tokens des signatures. Le modèle des CRF obtient le meilleur F-score : 79%.

3 Jeux de données

Deux jeux de données annotés ont été fournis aux participants pendant la phase de mise au point :

— le jeu de données réaliste (JDR), composé de 473 signatures, produit par la société Isahit en lien avec le comité d'organisation du défi ;

— le jeu de données factice (JDF), composé de 606 signatures créées automatiquement, par `fakenamegenerator.com`, une API de création de fausses identités. Certaines étiquettes ne sont cependant pas représentées.

Un jeu de données authentique (JDA), sans annotation, a été fourni pour valider notre système en fin de défi. Les signatures de ce jeu ont été recueillies par un formulaire Web permettant le "don" de signatures qui ont ensuite été pseudonymisées.

Les trois corpus ont été annotés avec le jeu de 13 étiquettes présenté en table 1. Les caractéristiques des trois jeux de données sont présentées en table 2.

Pour chaque signature, on trouve un identifiant `identifier`, un champ textuel `text` qui fournit le texte de la signature, sans mise en forme riche, ainsi que la liste des annotations à produire `annotations`. Pour chaque annotation attendue, on trouve la graphie `form`, son étiquette `label`, l'index `begin` de son premier caractère, l'index `end` du caractère suivant le token.

Dans les jeux de mise au point JDR et JDF le champ `label` est fourni. Dans le jeu de validation JDA, ce champ est absent puisqu'il doit être produit par le système automatique.

4 Méthode

Suivant la tradition du traitement automatique des langues et des documents de notre équipe, nous avons choisi de réaliser une chaîne de traitement complète, depuis la segmentation des signatures jusqu'à la production du fichier d'annotations au format attendu.

TAB. 1 – *Définition du jeu d'étiquettes*

Étiquette	Définition
Human	Identité des personnes qui figurent dans la signature
Organization	Organisation à laquelle l'auteur de la signature est rattachée
Function	Ensemble des fonctions assignées à la personne identifiée dans la signature
Project	Projet dans lequel la personne est impliquée
Location	Bâtiments, bureaux, villes, numéros et noms des rues
Reference_CEDEX	Courrier d'entreprise à distribution exceptionnelle
Reference_CS	Course spéciale
Reference_Code_Postal	Code postal
Phone_Number	Numéro de téléphone ou de fax
Email	Adresses e-mails
Url	URL de site web
Social_Network	Nom des réseaux sociaux
Reference_User	Identifiant d'une personne ou d'une organisation sur un réseau social

Contrairement aux approches automatiques reposant sur la catégorisation d'une suite de tokens, nous avons choisi de réaliser une segmentation des énoncés en constituants, puis de catégoriser chaque constituant. L'étiquetage des mots, attendu dans le défi, a alors consisté à projeter l'étiquette de chacun des constituants sur les mots qui le constituent.

Cette approche qui fait intervenir des unités intermédiaires nous semble particulièrement intéressantes puisqu'elle réduit considérablement le nombre d'unités à traiter et limite la taille de la fenêtre d'observation pour les déductions contextuelles basées sur les unités situées avant ou après l'unité traitée. Cette stratégie peut être vue comme une approche de type *diviser pour régner*.

Bien entendu, une segmentation en constituants logiques ne peut être obtenue d'emblée et une segmentation approximative, en constituants graphiques, est utilisée. Ainsi, nous utilisons deux classes de délimiteurs : les délimiteurs phrastiques de type "saut de ligne", et les délimiteurs intraphrastiques, de type "tiret", "tiret-long", "barre oblique", "puce".

Le typage des constituants graphiques ainsi obtenus est réalisé par analyse morphologique, et analyse contextuelle. Les étiquettes sont divisées en trois classes : les étiquettes attribuées à des contenus peu ambigus et ne nécessitant que peu d'analyse contextuelle (adresse e-mail, numéro de téléphone, nom de domaine, réseaux sociaux), les étiquettes qui pour être posées nécessitent des ressources linguistiques (adresse, fonction), et les étiquettes nécessitant une analyse contextuelle (identité, nom de la société, nom de projet).

Pour l'analyse non contextuelle, nous faisons usage d'expressions régulières, en particulier pour les numéros de téléphone, adresses e-mail, noms de domaine, réseaux

TAB. 2 – *Répartition des étiquettes dans les 3 jeux de données*

Étiquette	JDR		JDF		JDA	
Human	971	14,51%	1023	19,58%	1196	13,46%
Organization	1023	15,29%	943	18,05%	1537	17,29%
Location	2533	37,86%	2150	41,15%	2680	30,15%
Phone_Number	473	7,07%	371	7,10%	688	7,74%
Function	567	8,47%	0	0,00%	1449	16,30%
Email	297	4,44%	371	7,10%	344	3,87%
Url	227	3,39%	0	0,00%	303	3,41%
Social_Network	18	0,27%	0	0,00%	28	0,32%
Reference_User	9	0,13%	0	0,00%	11	0,12%
Reference_Code_Postal	269	4,02%	367	7,02%	349	3,93%
Project	98	1,46%	0	0,00%	124	1,40%
Reference_CEDEX	150	2,24%	0	0,00%	146	1,64%
Reference_CS	56	0,84%	0	0,00%	33	0,37%
	6691		5225		8888	

sociaux. Il en est de même pour l'identification d'adresses basée sur les codes postaux, CEDEX et CS. Pour l'identification des autres constituants adresse, nous avons recours à des ressources de types de voie communs, et à des attendus tels qu'un numéro de voie. Pour la fonction, nous utilisons un lexique de noms de métiers, généralisé, complété par des suffixes communs de noms de métiers (e.g., -ogue, -iste). Enfin, nous exploitons également des introducteurs caractéristiques de type de constituants tels que "Tél :", "Adresse :".

Pour l'analyse positionnelle, nous exploitons le fait que l'identité arrive en début de signature. Une analyse contextuelle permet d'attribuer le nom de société, considéré obligatoire, et les éventuels noms de projet, parmi les constituants non catégorisés, quitte à remettre en cause le pré-étiquetage non fiable d'un constituant. C'est également dans l'analyse contextuelle que sont exploités les noms de métiers et leurs suffixes fréquents.

Afin de fiabiliser les déductions sur les noms de personnes et les noms de société, nous effectuons un calcul des chaînes de coréférence, mettant en relation le nom de la personne et son identifiant d'e-mail, et le nom de la société avec les noms de site internet et noms de domaines de l'adresse e-mail. Ce processus à la fois original et efficace, est basé sur le calcul de distance d'édition de chaînes de caractère. Il n'est cependant pas utilisable sur le corpus de validation pseudonymisé.

Enfin, pour plonger dans le format attendu et de garantir la fiabilité de certaines déductions, un post-traitement est effectué et force le réétiquetage de certains tokens (CEDEX, e-mail, téléphone, nom de domaine). Notre segmentation en token est alignée avec la segmentation attendue.

5 Résultats

La performance de notre système est respectivement évaluée à 99% et 100% de f-mesure sur les jeux de mise au point réaliste (JDR) et factice (JDF), voir les tables 3 et 4. Sur le jeu de données authentique (JDA) utilisé pour la validation de notre système, la performance se dégrade de manière significative à 83% de f-mesure, avec une précision et un rappel sensiblement équivalents, voir table 5.

L'explication d'un tel écart de performance entre la phase de mise au point et la phase de validation pourrait être attribuée à une trop grande spécificité de notre système aux données de mise au point. S'arrêter à cette conclusion serait cependant hâtif. Les causes sont certainement multiples, mais il nous semble que la différence de nature des jeux de données de mise au point et de validation n'est pas neutre et mérite une attention toute particulière.

Les jeux de données JDR et JDF sont en effet artificiels et respectent des contraintes syntaxiques qui ont fait l'objet de spécifications (Cousot et al., 2023). Le jeu de validation JDA est quant à lui composé de données authentiques, sans contraintes de bonne formation. On observe par conséquent une variabilité formelle et lexicale bien plus importante dans ce dernier.

L'absence de marques de structuration dans le JDA n'est pas sans impact. Les données de ce jeu ont été récupérées par l'intermédiaire d'un formulaire de "don" de signature en ligne. Ce dispositif de captation, adapté pour les signatures purement textuelles, ne permet pas la récupération des informations structurelles du format HTML : les éléments de structuration (`div`, `br`, ...) sont ainsi perdus, engendrant des signatures composées parfois d'une unique ligne de texte, sans aucun séparateur exploitable. L'impact est significatif puisque ces marques, observables et exploitables dans la phase de mise au point pour délimiter les constituants et mettre en œuvre le critère positionnel, ne sont plus disponibles. Autre conséquence liée que nous attribuons au mode de captation : nous notons le doublement, voire le triplement de certains constituants de l'adresse dans le jeu de validation, ce qui a pour conséquence de rendre inexploitable le critère d'unicité de certains constituants.

La pseudonymisation appliquée sur le jeu de validation JDA est également à considérer dans la perte de performance. En remplaçant systématiquement et de manière aléatoire les noms de personnes, les noms d'organisation, les noms de domaine des adresses e-mail et des URLs, la procédure casse la cohérence textuelle des signatures, en particulier le schéma relationnel qui est exploitable dans les jeux de mise au point.

Enfin, la dimension multilingue représentée dans les jeux de mise au point JDR et JDF est absente du jeu de validation JDA qui n'est par exemple composé que d'adresse française.

Pour contrecarrer cette différence importante de forme entre les jeux de mise au point et de validation, nos efforts pendant la période de validation ont porté sur la relativisation des séparateurs de constituants, et sur leur délimitation basée sur des critères formels. Le temps limité de la phase de validation ne nous a cependant pas permis de mettre en œuvre l'ensemble des indices potentiellement exploitables.

6 Conclusion

Dans cet article, nous avons présenté notre participation au défi TextMine'23 (Cousot et al., 2023) "Reconnaissance d'entités d'intérêts dans les signatures d'e-mails".

Nous avons choisi de réaliser une chaîne de traitement complète, depuis la segmentation des signatures jusqu'à la production du fichier d'annotations au format attendu. La performance de notre système atteint 99% et 100% en f-mesure sur les jeux de données de mise au point et 83% de f-mesure sur le jeu de validation.

Parmi les options stratégiques que nous avons retenues, nous pensons que le fait de chercher à étiqueter non pas des tokens mais des unités de plus haut niveau est particulièrement différenciant. Nous pensons à ce titre qu'une évaluation basée sur la délimitation et l'étiquetage des constituants logiques serait pertinente et complémentaire à celle menée sur les tokens.

Afin de fiabiliser les déductions sur les noms de personnes et les noms de société, nous avons mis au point un calcul des chaînes de coréférence, mettant en relation le nom de la personne et son identifiant d'e-mail, et le nom de la société avec les noms de site internet et noms de domaines de l'adresse e-mail.

La technique d'analyse utilisée a l'intérêt de fournir des résultats explicables et interprétables.

La chute de performances entre la phase de mise au point et la phase de validation a été longuement analysée. Nous avons mis en évidence les limites du jeu de données authentique. Nous pensons cependant que le jeu de données authentique (JDA) est très certainement celui qui présente le plus d'intérêt puisqu'il propose des données réelles. Le processus d'acquisition devrait cependant être revu pour permettre la préservation des informations structurelles. Le processus de pseudonymisation devrait quant à lui également être reconsidéré pour conserver la cohérence textuelle.

Références

Bendahman, N., K. Cousot, et C. Lopez (2022). Reconnaissance d'entités d'intérêt dans les signatures d'e-mails à partir d'un jeu de données authentique. *TextMine'22*.

Carvalho, V. R. et W. W. Cohen (2004). Learning to extract signature and reply lines from email. In *Proceedings of the Conference on Email and Anti-Spam*, Volume 2004.

Chen, H., J. Hu, et R. W. Sproat (1999). Integrating geometrical and linguistic analysis for email signature block parsing. *ACM Transactions on Information Systems (TOIS) 17*(4), 343–366.

Cousot, K., C. Lopez, P. Cuxac, et V. Lemaire (2023). Défi textmine'23 - reconnaissance d'entités d'intérêts dans les signatures d'e-mails. In *Actes de l'atelier TextMine'23, Conférence Extraction et Gestion des Connaissances 2023 (EGC'23)*, Lyon, pp. à paraître.

Estival, D. (2008). Author attribution with email messages. *Journal of Science, Vietnam National University 1*, 1–9.

Giguet, E. et G. Lejeune (2021). Daniel at the FinSBD-2 task : Extracting list and sentence boundaries from PDF documents, a model-driven approach to PDF document analysis. In *Proceedings of the Second Workshop on Financial Technology and Natural Language Processing*, Kyoto, Japan, pp. 67–74. -.

Giguet, E. et N. Lucas (2010). The book structure extraction competition with the resurgence software at caen university. In S. Geva, J. Kamps, et A. Trotman (Eds.), *Focused Retrieval and Evaluation*, Berlin, Heidelberg, pp. 170–178. Springer Berlin Heidelberg.

Giguet, E. et N. Lucas (2022). GREYC@FinTOC-2022 : Handling Document Layout and Structure in Native PDF Bundle of Documents. In M. El-Haj, P. Rayson, et N. Zmandar (Eds.), *4th Financial Narrative Processing Workshop (FNP 2022)*, Marseille, France, pp. 100–104.

Li, J., S. Sen, et N. Zaman (2015). Entity extraction from business emails. *International Journal of Information Technology and Computer Science 7*(9), 15–22.

Summary

This paper presents our contribution to the TextMine'23 Challenge related to "Named Entity Recognition in Email Signatures". The performance of our system reaches 99% and 100% F1-score on the training sets and 83% F1-score on the test set.

TAB. 3 – *Évaluation sur le jeu de mise au point JDR*

	précision	rappel	f1-score	support
email	1.00	1.00	1.00	297
function	0.99	0.99	0.99	567
human	1.00	1.00	1.00	971
location	1.00	1.00	1.00	2538
organization	0.98	0.98	0.98	1018
phone_number	1.00	0.99	0.99	473
project	0.87	0.84	0.85	98
reference_cedex	0.96	1.00	0.98	150
reference_code_postal	1.00	1.00	1.00	269
reference_cs	1.00	1.00	1.00	56
reference_user	1.00	1.00	1.00	9
social_network	1.00	1.00	1.00	18
url	1.00	1.00	1.00	227
micro avg				
macro avg	0.98	0.98	0.98	6691
weighted avg	0.99	0.99	0.99	6691
F1	0.9912928493115661			

TAB. 4 – *Évaluation sur le jeu de mise au point JDF*

	précision	rappel	f1-score	support
email	1.00	1.00	1.00	371
function	0.00	0.00	0.00	0
human	1.00	1.00	1.00	1023
location	1.00	1.00	1.00	2150
organization	1.00	1.00	1.00	943
phone_number	1.00	1.00	1.00	371
project	0.00	0.00	0.00	0
reference_cedex	0.00	0.00	0.00	0
reference_code_postal	1.00	1.00	1.00	367
reference_cs	0.00	0.00	0.00	0
reference_user	0.00	0.00	0.00	0
social_network	0.00	0.00	0.00	0
url	0.00	0.00	0.00	0
micro avg	1.00	1.00	1.00	5225
macro avg	0.46	0.46	0.46	5225
weighted avg	1.00	1.00	1.00	5225
F1	0.9997124612816258			

TAB. 5 – *Évaluation sur le jeu de validation JDA*

	précision	rappel	f1-score	support
email	1.0000	1.0000	1.0000	344
function	0.8881	0.7723	0.8261	1449
human	0.8582	0.9055	0.8812	1196
location	0.9101	0.8540	0.8811	2678
organization	0.6023	0.6415	0.6213	1537
phone_number	0.9663	1.0000	0.9829	688
project	0.0762	0.1935	0.1093	124
reference_cedex	0.9799	1.0000	0.9898	146
reference_code_postal	0.8876	0.9107	0.8990	347
reference_cs	0.0000	0.0000	0.0000	37
reference_user	0.4000	0.1818	0.2500	11
social_network	0.9310	0.9643	0.9474	28
url	0.9967	1.0000	0.9984	303
micro avg	0.8243	0.8241	0.8242	8888
macro avg	0.7305	0.7249	0.7220	8888
weighted avg	0.8414	0.8241	0.8312	8888
F1	0.8311914814126119			

Hybridation des approches symboliques et apprentissage profond pour la reconnaissance des entités dans les signatures de mail

Duc Hau Nguyen*, Nicolas Fouqué*, Victor Klötzer**, Hugo Thomas*

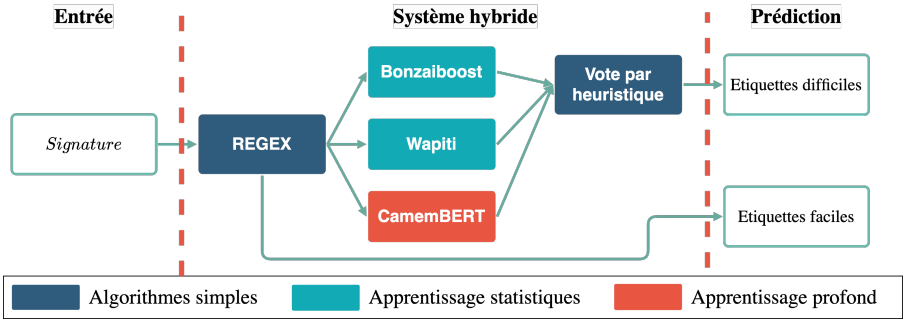
* IRISA, CNRS, INSA Rennes, Université de Rennes
Prenom.Nom@irisa.fr
** INRIA, Université de Rennes
victor.klotzer@inria.fr

1 Reconnaissances des entités dans des signature de mail : défi Textmine'23

Le défi Textmine'23 (Cousot et al. (2023)) porte sur la classification d'entités parmi 12 classes à l'intérieur de signatures de mails. Ces entités sont déjà identifiées dans les données fournies. Le défi présente 3 jeux de données : **Jeu de données réaliste** (JDR) [1], **Jeu de données factice** (JDF) [2] et **Jeu de données authentique** (JDA) [3]. Le JDA sert de jeu d'évaluation, pour les modèles entrainés sur le JDR et le JDF.

2 Système hybride de symbolique et apprentissage

FIG. 1 – *Le système de Reconnaissance des Entités Nommées en prenant en compte 4 modèles*



Pour un meilleur compromis entre performance et robustesse, notre système consiste en 4 modules de reconnaissance des entités nommées, entraînés indépendamment :

1. des signatures construite manuellement par Isahit : https ://fr.isahit.com
2. des signatures générée automatiquement par https ://www.fakenamegenerator.com
3. composées des signatures authentiques pseudonymisées

— **Expressions régulières** ou *RegEx* (Aho (1991)) : une expression régulière est un motif qui décrit un ensemble de chaînes de caractères possibles que l'on veut repérer dans une autre chaîne de caractères plus longue. L'utilisation de RegEx est très efficace pour identifier les entités ayant des structures simples et toujours similaires : numéros de téléphone, codes postaux, adresses mail, URLs, cedex, réseaux sociaux et réf. user.
— **Bonzaiboost** (Laurent et al. (2014)) : un modèle repose sur le boosting d'arbres de décisions peu profonds (les bonsaïs). La décision du modèle (ici unigramme) est basé sur (1) la présence et l'absence de certain *tokens* et (2) des connaissances extérieures formulées sous la forme de RegEx exploitées en booléens. À l'entraînement, on utilise un paramètre de sur-échantillonage pour les classes sous-représentées dans les données d'apprentissage.
— **Wapiti (CRF)** (Lavergne et al. (2010) : les modèles CRF (ou champs aléatoires conditionnels) sont une famille de modèles discriminatifs qui représentent un réseau probabiliste non-orienté : le modèle appris peut être considéré comme un graphe de noeuds-états entre lesquels les arrêtes sont munies de probabilités conditionnelles de transition. Un CRF (dont l'outil Wapiti possède une implémentation) peut donc prendre en compte les variables voisines et modéliser des séquences, d'où son utilisation pour des tâches d'étiquetage de séquence.
— **CamemBERT** (Martin et al. (2020)) : CamemBERT est un modèle de type Transformers pré-entraîné sur des corpora français pour de multiples tâches. Nous exploitons ce modèle en le spécialisant sur la tâche de reconnaissance des entités nommées.

Notre système se repose en priorité sur les RegEx pour identifier les étiquettes "simples" (repérable indépendamment du contexte). Pour les entités qui ne correspondent à aucune RegEx, on procède à un vote à partir des prédictions des trois autres modèles (**Bonzaiboost**, **Wapiti** et **CamemBERT**).

Dans le cas où les trois modèles sont en désaccord, un ordre de priorité (déterminé empiriquement) a été attribué aux modèles : la prédiction de **CamemBERT** est utilisée, excepté dans le cas où **CamemBERT** n'a prédit aucune classe, auquel cas nous utilisons la prédiction de **Wapiti**, et en dernier recours la prédiction de **Bonzaiboost** (configuré pour toujours donner une prédiction).

3 Résultats et conclusion

Composants	F-score	Précision	Rappel
RegEx + Wapiti + Bonzaiboost	0.757	0.785	0.680
RegEx + Wapiti + Bonzaiboost + CamemBERT + vote	0.807	0.835	0.714
RegEx + Wapiti + Bonzaiboost + CamemBERT + vote avec heuristique de priorisation	0.832	0.800	0.778

Le système sans heuristique de priorisation a tendance à rater des entités. Les classes sur lesquelles le modèle est le moins performant sont essentiellement les classes sous représentées et qui ne sont pas détectables par RegEx. L'heuristique de priorisation permet d'étiqueter plus d'entités en prenant le risque d'accepter des prédictions à tort (gain 0.06 en rappel et perte 0.03 en précision) ; un compromis qui permet d'améliorer la performance globale du système.

Références

Aho, A. V. (1991). Algorithms for finding patterns in strings, handbook of theoretical computer science (vol. a) : algorithms and complexity.

Cousot, K., C. Lopez, P. Cuxac, et V. Lemaire (2023). Défi textmine'23 - reconnaissance d'entités d'intérêts dans les signatures d'e-mails. In *Interspeech*, Lyon. Conférence Extraction et Gestion des Connaissances 2023 (EGC'23).

Laurent, A., N. Camelin, et C. Raymond (2014). Boosting bonsai trees for efficient features combination : application to speaker role identification. In *Interspeech*, Singapour.

Lavergne, T., O. Cappé, et F. Yvon (2010). Practical very large scale CRFs. In *Proceedings the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 504–513. Association for Computational Linguistics.

Martin, L., B. Muller, P. J. Ortiz Suárez, Y. Dupont, L. Romary, É. de la Clergerie, D. Seddah, et B. Sagot (2020). CamemBERT : a tasty French language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online, pp. 7203–7219. Association for Computational Linguistics.

Summary

Deep learning has met success recently in natural language processing, especially with Transformers. However, these methods meet limitation when data is limited and evaluation data introduces more noise than training data. To overcome this problem, we suggest a hybridization technique that takes into account CamemBERT model with two traditional machine learning techniques Bonzaiboost (decision trees boosting) and Wapiti (CRF) as well as simple regular expressions. Our technique manifests good results to overcome limitations through the task that is presented as the Textmine'23 challenge.

Nextino@TextMine'23 : Approche hybride pour la reconnaissance d'entités d'intérêt dans les signatures d'e-mail

Maëlle Brassier*,** Asceline Goudjo**

* LIFAT, Université de Tours, 41000 Blois, France
prenom.nom@etu.univ-tours.fr
**Nextino, 1 Avenue du Champ de Mars, 45100 Orléans, France
prenom.nom@nextino.eu

Résumé. Cet article présente le système développé par l'équipe TAL de Nextino pour le défi TextMine'23 portant sur la classification des entités d'intérêt dans les signatures d'e-mails. Nous proposons une approche par hybridation qui allie les forces de différentes méthodes - expressions régulières, CRF et Transformers - pour faire face aux spécificités de chaque type d'entité d'intérêt. Cette approche est complétée par une étape de post-traitements pour améliorer la classification des entités pour lesquelles nous avons rencontré le plus de difficultés. De plus, nous avons cherché à répondre à la fois à la problématique de classification proposée par ce défi mais également à la problématique sous-jacente de détection des entités en plein texte. Nous avons soumis 3 runs et obtenu la première place à ce défi avec un F-score de 0.908.

1 Introduction

La Reconnaissance d'Entités Nommées (REN) est une tâche du Traitement Automatique des Langues qui, au fil des années, a pu être appliquée dans de nombreux domaines. Néanmoins, face à des problématiques de confidentialité et de respect du droit à la vie privée, peu de cas applicatifs ont pu être explorés, éloignant alors les données de situations réelles. Le défi TextMine'23. propose un contexte d'application sur données réelles à travers une tâche de reconnaissance d'entités d'intérêt dans des signatures d'e-mails. Les entités étant déjà détectées et disponibles, la tâche se traduit comme un problème de classification de tokens à 13 classes, où chaque entité correspond à une donnée reliée à une personne physique.

L'équipe TAL de Nextino s'intéresse à la détection des données personnelles dans un objectif final d'anonymisation. Tirant profit de la similitude des données et de la tâche, notre équipe a proposé une approche hybride composée d'un ensemble de modules, chacun dédié à une entité d'intérêt. Cette hybridation allie trois approches : approche symbolique, statistique et neuronale et nous a permis de nous rapprocher le plus possible des caractéristiques de chacune d'entre elles ainsi que d'obtenir les meilleures performances du défi.

Dans cet article, nous dresserons dans un premier temps un état de l'art sur la tâche de Reconnaissance d'Entités Nommées (section 2), puis nous présenterons nos expérimentations,

parallèlement aux données (sous-section 3.1), approches (sous-section 3.2) et étape de post-traitements (sous-section 3.3. Enfin, nous discuterons des résultats obtenus ainsi que les perspectives d'amélioration (section 4).

2 État de l'art

Souvent présente en début de chaîne de traitements ou de pipeline d'applications de TAL, la Reconnaissance d'Entités Nommées est une tâche d'extraction d'information consistant à identifier des expressions linguistiques afin de les catégoriser dans des classes définies. Ces entités se réfèrent généralement à des noms propres tels que des noms de personnes, d'organisations, de lieux mais peuvent également concerner des dates, des quantités etc. À l'instar de nombreuses applications, les premiers travaux de la REN se sont reposés sur des ensembles de règles, elles-même basées sur des ressources externes tels que des dictionnaires, des lexiques ou ontologies. Afin de s'affranchir d'un travail conséquent de création de règles manuelles, des algorithmes de Machine Learning ont été appliqués pour la tâche, tels que des arbres de décisions, des HMM (Hidden Markov Models), des SVM (Support Vector Machines) ou encore des CRF (Condition Random Field). Shengyu et al. (2015) ont dépassé le système gagnant de la campagne d'évaluation DDIExtraction 2013 challenge (Segura-Bedmar et al., 2013) en proposant un système à base de CRF qui combine l'utilisation de features provenant de lexiques et de word embeddings appris sur le corpus biomédical MEDLINE. Enfin, face à l'émergence de l'apprentissage profond, les travaux les plus récents de la REN se sont focalisés majoritairement sur la représentation des données et donc des entités ainsi que sur le perfectionnement des mécanismes de l'architecture Transformer. Li et al. (2018) dressent un état de l'art complet de ces différents travaux et démontrent la supériorité de ces approches en termes de performances mais également leur dépendance à de larges corpus annotés.

La Reconnaissance d'Entités Nommées se retrouve dans de nombreux domaines, notamment le domaine médical sous la déclinaison de la tâche de dé-identification d'informations de santé protégées (PHI pour Protected Health Information) dans des textes médicaux. Les systèmes proposés ont suivi la même trajectoire, avec des premiers travaux symboliques (Sweeney, 1996), suivis par des méthodes statistiques (Guo et al., 2006). Les travaux les plus récents prônent des approches hybrides, combinant des réseaux de neurones avec des CRF (Liu et al., 2017) ou des approches à base de règles telles que la structure des documents ou les patterns des formes de surface des entités. Outre le domaine (bio-)médical, la plupart des données des corpus de REN proviennent de journaux de presse, tel que le dataset CoNLL2003 (Tjong Kim Sang et De Meulder, 2003) dont la source est le corpus Reuters News ou encore le dataset CBS SciTech News (Jia et al., 2019). D'autres datasets proposent des sources aux domaines différents, comme le dataset Email (Lawson et al., 2010) et le dataset Twitter (Derczynski et al., 2016). Néanmoins, aucun de ces datasets ne se rapprochent d'un contexte industriel et professionnel, dans lequel se trouvent des informations personnelles liées à un individu.

3 Expérimentaiton

3.1 Les données

Les organisateurs du défi ont mis à disposition des participants trois jeux de données annotés contenant uniquement des signatures produites selon différentes stratégies. Les données sont annotées à partir des 13 classes présentées dans le tableau 1.

Classe	Définition
Human	Noms et prénoms des personnes qui figurent dans la signature
Organization	Organisation à laquelle l'auteur de la signature est rattachée
Function	L'ensemble des fonctions assignées à la personne identifiée
Project	Projet dans lequel la personne est impliquée
Location	Bâtiments, bureaux, villes, numéros et noms des rues
Reference_CEDEX	Courrier d'entreprise à distribution exceptionnelle
Reference_CS	Course spéciale
Reference_Code_Postal	Code postal
Phone_Number	Numéro de télphone ou de fax
Email	Adresses e-mails
Url	URL vers le site web de l'entreprise ou la personne
Social_Network	Nom des réseaux sociaux
Reference_User	Identifiant sur un réseau social

TAB. 1 – *Classes annotées dans les jeux de données (Cousot et al., 2022)*

— Le **JDA** (Jeu de Données Authentique) : jeu de données composé de 606 vraies signatures collectées par un formulaire partagé via des listes de diffusion à destination majoritairement de français (Bendahman et al., 2022). Ces données ont été pseudonymisées mais selon nos observations sans conservation des liens entre les données. (exemple : "11100 Montpellier" : le code postal ne correspond pas à celui de la ville.)
— le **JDR** (Jeu de Données Réaliste) : jeu de données composé de 473 signatures produites par la société Isahit selon un cahier des charges suivant les spécificités du JDA.
— le **JDF** (Jeu de Donnés Factice) : jeu de données composé de 500 signatures créées automatiquement. La génération s'appuie sur différentes règles et une API capable de générer de fausses identités.

Chaque jeu de données est présenté dans un fichier JSON illustré dans la Fig 1. Dans ces données, les entités sont directement identifiées, rapprochant cette tâche d'une tâche de classification plutôt que de détection. L'évaluation finale du défi s'est effectuée sur le JDA qui a été mis à disposition à la fin du challenge. Pour travailler sur des jeux de données équilibrés, nous avons mélangé de manière aléatoire le JDF et le JDR et nous avons scindé l'ensemble des données en 3 sous corpus : *Train*/*Dev*/*Test* avec une répartition de respectivement 60/20/20. Ce découpage a servi notamment aux modules se reposant sur des approches à base d'apprentissage. Tous les modèles ont été entraînés exclusivement sur le *Train* et les premières évaluations ont été réalisées sur le *Dev*. Le corpus *Test* a été utilisé pour la phase d'évaluation.

3.2 Modules de détection : les approches

Chaque classe d'entités fait l'objet d'un module de détection distinct. Cette décision est motivée à la fois par notre expérience sur le traitement des données personnelles et à la fois par l'hétérogénéité des données à détecter. En effet, pour détecter ces entités il est possible de s'appuyer sur un certain nombre d'indices qui diffèrent selon la nature des données. McDonald (1993) les classe en deux catégories : les indices internes et les indices externes. Certaines entités possèdent des indices internes qui permettent d'identifier une entité se basant uniquement sur l'ensemble des caractères qui la composent et sur sa forme graphique. D'autres entités présentent des indices externes qui nécessitent de se baser sur le contexte dans lequel elles apparaissent pour les détecter.

Dans cette section nous présentons les différentes approches utilisées pour détecter et catégoriser les différentes classes d'entités d'intérêt. Ces modules prennent en entrée le texte de la signature. Ils effectuent donc une tâche de **détection** des entités en plein texte mais également de **classification** dont la classe découle intrinsèquement du module qui l'a détectée.

Expressions régulières : approche symbolique Plusieurs classes d'entités emploient des modules de détection qui font appel à des approches symboliques. Ces entités sont celles qui présentent le plus d'indices internes. Il s'agit d'indices graphiques comme la présence de chiffres ; d'indices morpho-syntaxiques qui permettent d'inférer sa catégorie (e.g verbe) ou encore d'indices sur les caractères spéciaux ou la ponctuation. Ce type d'indice permet d'identifier les entités qui ont une forme standardisée et/ou des caractères spécifiques. C'est le cas des entités *Phone_Number*, *Email* et *Url*. L'objectif de l'approche symbolique est de se rapprocher des données et de leur représentation en créant un ensemble de règles capable de les détecter de la manière la plus exhaustive possible. Ces règles ont été écrites à l'aide d'expressions régulières.

- Le module de détection des *Email* s'appuie sur le modèle de Reconnaissance d'Entités Nommées de Spacy (Honnibal et al., 2020) pour détecter une partie des adresses e-mail. À la suite de ce premier traitement, des expressions régulières sont appliquées pour détecter les adresses plus complexes (e.g adresses contenant des caractères accentués).
- Les modules de détection des *Phone_Number* et des *Url* ont recours uniquement à des expressions régulières en s'appuyant sur l'homogénéité de leur forme pour décrire toutes les combinaisons de caractères possibles.
- Enfin, le module de détection des *Social_Network* repose sur la simple détection d'une liste de réseaux sociaux dans la signature sans prise en compte de la casse.

CRF : approche statistique Contrairement aux approches symboliques qui tentent de capturer des entités à partir d'un ensemble de règles le plus exhaustif possible, les approches statistiques vont davantage se reposer sur le principe d'un apprentissage automatique, où le modèle va apprendre à partir des données. Nous utilisons ici uniquement la méthode de CRF Lafferty et al. (2001), une méthode statistique qui se repose sur une approche conditionnelle pour représenter et étiqueter une séquence de données. Cette méthode s'applique donc particulièrement à des données qui vont posséder un caractère séquentiel et qui vont également dépendre de leur contexte avoisinant, à savoir toutes les entités de type localisation (*Location*, *Reference_Cedex*, *Reference_CS*, *Reference_Code_Postal*), *Reference_User* et *Project*.

— Toutes les entités de type localisation ont été regroupées dans un seul module. Puisque les adresses postales suivent naturellement une séquence de sous-adresses les CRF ont pu tirer parti de ce caractère séquentiel où l'ordre de chaque token influence sur la probabilité des tokens voisins (e.g un numéro d'adresse précède généralement un nom de rue et un nom de ville succède un code postal). Chaque token est également décrit par un ensemble de traits, tels que la taille du token, des booléens sur la nature du token (i.e s'il s'agit d'un nombre, d'un type de voierie, d'une ville) ainsi que les traits des tokens voisins (sur une fenêtre de 2). Le modèle CRF va donc effectuer une classification multi-classes de ces 4 classes.
— Les modules *Project* et *Reference_User* reposent sur le même principe bien que *Reference_User* soit moins dépendant de son voisinage mais davantage sur des indices internes qui peuvent être modélisés en tant que traits tels que l'inclusion d'un réseau social dans le token, si le token commence par un @ ou s'il s'agit d'une URL.
— Le module *Project*, quant à lui, repose sur un ensemble de traits plus restreint comme la détection de trigger words propres aux projets (e.g "Licence", "Département" etc).

Transformer : approche neuronale Certaines entités ne peuvent être décrites à base de règles car elles ne présentent pas d'indice interne et ne peuvent de ce fait pas s'appuyer sur un caractère séquentiel qui est inexistant. Il est alors nécessaire de s'appuyer sur d'autres indices externes, notamment le contexte dans lequel elles apparaissent pour en extraire une dimension sémantique. C'est le cas des entités *Human*, *Organization* et *Function*. Les modules de détection de ces entités se basent sur des Transformers (Vaswani et al., 2017). Cette architecture présente deux avantages. Le premier est la capacité à effectuer des calculs en parallèle et donc d'accélérer la phase d'entraînement, contrairement aux Réseaux de Neurones Récurrents. Le deuxième consiste à dupliquer le mécanisme d'attention avec le Multi-Head, permettant ainsi de capter différents types de relations entre tokens (grammaticales, de proximité...) pour se focaliser seulement sur les éléments pertinents (Voita et al., 2019). Deux modèles basés sur des Transformer ont été utilisés : la version multilingue de BERT (appelée ici BertMulti) entraînée sur le contenu de Wikipédia dans les 104 langues les plus importantes (Devlin et al., 2018) et CamemBERT (Martin et al., 2020), une version française de BERT.

— Pour les modules de détection des *Human* et *Organization*, nous avons adopté le modèle BertMulti. L'observation des données a révélé le caractère multilingues de certaines informations comme le nom des organisations. Les hyper-paramètres des modèles *Organization* et *Human* sont de, respectivement : une taille de batch de 35 et 7, un nombre de 3 epochs, un learning rate fixé à 5e-05 et 3e-05 et enfin une longueur maximale de séquence de 128 et 512.
  Pour améliorer les performances, nous avons rajouté une étape de pre-traitements. Une première fonction a pour objectif d'harmoniser et/ou supprimer les espaces et les séparateurs en début et fin de signature. La deuxième fonction permet de marquer les retours à la ligne dans la signature en les remplaçant par un token spécial "µ" car par défaut, si les données en entrée ne sont pas données phrase par phrase au tokenizer de BERT, celui-ci ne conserve pas l'information du saut de ligne. Le caractère très structuré des signatures nous a poussé à rajouté cette feature.
— Du côté des *Function*, le modèle CamemBERT a été implémenté en partant du postulat que les données sont constituées majoritairement de signatures en français. Les

Listing 1: Données de test

```
[
  {
    "identifier" : 0,
    "text" : "Faustin Dupont",
    "annotations" : [
      {
        "form" : "Faustin",

        "begin" : 0,
        "end" : 7
      },
      {
        "form" : "Dupont",

        "begin" : 8,
        "end" : 14
      }
    ]
  }
]
```

Listing 2: Soumission du participant

```
[
  {
    "identifier" : 0,
    "text" : "Faustin Dupont",
    "annotations" : [
      {
        "form" : "Faustin",
        "label" : "Human",
        "begin" : 0,
        "end" : 7
      },
      {
        "form" : "Dupont",
        "label" : ""     ,
        "begin" : 8,
        "end" : 14
      }
    ]
  }
]
```

FIG. 1 – *Exemple JDA avec label manquant pour "Dupont" (Cousot et al., 2022)*

hyperparamètres sont les suivants : une taille de batch de 10, un nombre de 3 epochs, un learning rate fixé à 3e-05 et une longueur maximale de séquence de 75.

3.3 Modules de classification : post-traitements

Les modules de post-traitement se décomposent en deux catégories :

— Module de traitement des multi-prédiction : un module qui traite les cas où une entité a obtenu plusieurs prédictions et donc plusieurs classes et qui nécessite des règles supplémentaires pour attribuer la classe finale.

— Module de règles par entité : un module de classification des entités du JDA qui n'ont obtenu aucune prédiction après l'application des modules de détection.

Ces modules prennent tous en entrée les valeurs des champs "*annotations*" du JSON pré-rempli avec le label identifié par les modules de détection. Ces modules vont s'intéresser aux entités dont le label est vide comme pour la "*form*" "Dupont" dans la Fig. 1. Pour plus de clarté, le JSON pré-rempli avec les entités identifiées et catégorisées par les modules de détection sera appelé *JDA-participant* dans le reste de l'article et les entités qui n'ont pas encore de label, *entité-vide*.

Module de traitement des multi-prédiction Ce module prend en entrée à la fois les champs "*annotations*" du *JDA-participant* et les prédictions des modules de détection. Il gère les cas où plusieurs modules fournissent une prédiction pour la même entité, puisque chaque module renvoie ses propres prédictions, indépendamment des autres. Ce cas sera nommé *multi-prédiction*

dans le reste de l'article. Un certain nombre de règles permet de prioriser une classe parmi les classes candidates.

Exemple : une des règles appliquées consiste à attribuer la classe du token précédant le token à l'*entité-vide*, si cette classe fait partie de l'ensemble des classes candidates. Dans la Fig. 1, l'entité "Dupont" est prédite comme *Human*, *Location* et *Organization*. Puisque, l'entité précédente "Faustin" est catégorisée comme *Human*, alors le label *Human* sera attribué à "Dupont".

Module de règles par entité Ce module prend en entrée à la fois les champs "*annotations*" du *JDA-participant* et les champs "*text*" pour avoir accès au texte complet des signatures. Il s'intéresse uniquement aux entités du *JDA-participant* qui ont n'ont aucun label à ce stade du processus. Ce module rajoute des règles pour pallier les failles des modules à base d'approches neuronales ou statistiques. Nous détaillons dans cette section quelques exemples de règles que nous avons implémentées.

Pour le **module *Human***, si une entité identique à l'*entité-vide* a déjà été détectée et catégorisée *Human* dans la signature, alors le label *Human* est attribué à l'*entité-vide*.

"Ganelon[Human] *Grenier[Human]* (Remplaçante de Ganelon[Human] **Grenier** durant son congé maternité)
Assistante de Ganelon[Human] **Grenier**
Ligne directe : 01.64.12.68.85"

Dans cet exemple, "Grenier" a déjà été détecté par le modèle Bert-Multi en tant que *Human* dans sa première occurrence contrairement aux deuxième et troisième occurrences. Les règles de post-traitement permettent d'attribuer la classe *Human* à autres occurrences de "Grenier".

Pour le **module *Location***, si l'entité suivant l'*entité-vide* a pour label *Location* et que l'*entité-vide* est un digit, alors le label *Location* lui est attribué si ces deux entités ne sont pas séparées de plus de trois caractères. Des règles similaires existent pour les *Reference_CEDEX* ou *Reference_Code_Postal*.

"Danielle LaGrande
Responsable de travaux
+33.07.66.87.45.35 | DanielleLaGrande@armyspy.com
decofor-logo-plomberie
PLOMBERIE-CHAUFFAGE-COUVERTURE DEPUIS 1972
Tél. +33 (0)1 88 14 64 70
Jeans Unlimited
167 *bis[Location]*,[Location] Rue[Location] de[Location][...]"

Dans la signature ci-dessus, "167" est une *entité-vide* de type digit suivie d'une entité catégorisée *Location*. Les règle de post-traitement permettent d'attribuer la classe *Location* à l'entité "167".

Pour le **module *Function***, nous utilisons également le texte de la signature pour prendre en compte les retours à la ligne. Nous avons émis l'hypothèse qu'une signature est structurée de manière à présenter un type d'information par ligne si aucun séparateur n'est présent sur

cette dite ligne. Ainsi, nous supposons que toutes les informations contenues sur une même ligne appartiennent à une même classe. Si la première entité de la ligne a été détectée comme *Function* ou appartient à une liste de professions [1] alors les *entités-vides* de cette même ligne sont catégorisées comme *Function*.

"*Directeur[Function]* du[Function] **TIC MARIND**
Bâtiment Jean Braconnier
21, Avenue Claude Bernard
49300 VILLEURBANNE cedex
Bureau : 245
Téléphone : 04 23 70 05 43
Fax : 05 64 68 69 30
Mèl : MethenaBergeron@rhyta.com
Web : BowlingExpo.fr"

Dans cette signature, "TIC" et "MARIND" sont des *entités-vides*. Les règles de post-traitement permettent d'attribuer la classe *Function* à ces entités.

4 Résultats

L'évaluation du défi s'est faite via des mesures classiques de rappel, précision et f-score sur le jeu de données JDA. Chaque équipe avait la possibilité de soumettre trois runs. Chacun de nos runs correspond à un modèle qui suit l'architecture suivante : chaque module prend en entrée le texte de chaque signature et envoie leur prédiction de façon indépendante, puis l'ensemble de ces prédictions est soumis à une étape de post-traitements de la sous-section 3.3. L'amélioration des modèles s'est faite de façon incrémentale, en se concentrant sur les entités d'intérêt affichant les performances les plus basses et généralement plus difficiles à détecter par nos modules, notamment les *Function* et les *Project*.

La Table 2 résume les différents résultats obtenus. On observe une augmentation de + 0.0771 en rappel entre le *run1* et le *run3*, qui peut être expliquée par l'ajout de règles de post-traitement qui ont pour but de récupérer les tokens sans prédiction ou bien corriger des erreurs potentielles, en fonction des tokens voisins et leurs prédictions. Néanmoins, ces règles partent d'une hypothèse et d'un postulat forts, qui ne peuvent pas être vérifiables à posteriori et de ce fait, dégradent la précision de -0.0103 points.

	précision	rappel	f-score
run1	0.9270	0.8294	0.8695
run2	0.9011	0.8906	0.8930
run3	0.9167	0.9065	0.9078

TAB. 2 – *Résultats des runs en termes de précision, rappel et f-score*

1. `http://www.nurykabe.com/dump/text/lists/liste%20de%20m%c3%a9tiers%20(fr).txt`

La Table 3 détaille les résultats obtenus, par type d'entité d'intérêt pour le *run3*. Les entités tombant sous l'approche symbolique (i.e *Phone_Number*, *Social_Network*, *Email* et *Url*) affichent les meilleurs résultats, avec les deux derniers obtenant un f-score de 1.000. Ces résultats élevés démontrent qu'une approche symbolique, lorsqu'elle couvre le plus de cas possibles, peut offrir des performances supérieures à des méthodes plus élaborées mais plus coûteuses. L'approche statistique par CRF qui englobe les entités de type *Location* et *Reference_User* présentent des résultats également élevés, bien qu'en deçà. Il est néanmoins important de noter que les entités *Project* et *Reference_CS* obtiennent un f-score de respectivement 0.3554 et 0.1026. La première entité s'explique par un manque de détails permettant de clarifier la définition même de ce que compose un projet et parfois une juxtaposition avec les entités de type *Organization*. Bien que l'entité *Reference_CS* ait une précision de 1.000, son rappel chute à 0.0541, dû à un manque d'indices permettant de les différencier d'un code postal. Enfin, *Function*, *Human* et *Organization* qui reposent sur une architecture neuronale ont des résultats également légèrement en-dessous mais plus que satisfaisants, compte tenu de leur difficulté relative à identifier.

Avec un f-score final de 0.9078, notre équipe a terminé 1ère de ce défi TextMine'23. Notre stratégie hybride, qui s'est concentrée sur la recherche d'une approche la plus adaptée à chaque type d'entité, s'avère donc être une solution offrant des performances satisfaisantes, construite sur une architecture modulaire favorisant l'explicabilité de ses modules. En termes d'amélioration, le module *Function* qui utilise le modèle de langage CamemBERT aurait pu être remplacé par le modèle BertMulti qui aurait pu tirer profit de la nature multilingue des fonctions. En outre, la catégorie *Project* qui fait appel à la méthode de CRF aurait également pu bénéficier d'une approche neuronale, étant donné l'identification complexe de ces données. Par ailleurs, nous avons identifié assez tard l'intérêt de considérer les retours à la ligne comme des séparateurs de type d'information. De plus, nous n'avons pas pris en compte l'ordonnancement des informations présentes dans les signatures (ex : les entités *Human* apparaissent plutôt sur la première ligne, les entités de type *Location* se répartissent souvent sur plusieurs lignes et commencent fréquemment par un digit - le numéro de la rue ou le code postal). Ces informations auraient pu constituer des features de nos modèles CRF. Du côté des *Project*, le peu de données d'entraînement ne nous a pas permis de définir correctement la catégorie. A la lecture du JDA, nous parvenions avec peine à distinguer les *Project* des *Function* ou des *Organization* en tant qu'observateur humain. Concernant les *Reference_CS*, une pseudonymisation des entités en gardant la cohérence entre le nom de la ville et le code postal nous aurait permis de les différencier plus facilement.

5 Conclusion

Nous avons présenté dans cet article la participation de l'équipe TAL de Nextino au Défi TextMine'23. Cette tâche s'est intéressée à la classification des entités d'intérêt dans les signatures d'e-mail. 5 équipes ont participé au challenge avec la possibilité de soumettre trois fichiers de prédictions. Nous avons proposé une approche par hybridation qui permet de proposer une stratégie adaptée aux spécificités de chaque type d'entité d'intérêt. En effet, les précédents travaux sur la dé-identification dans le domaine médical et notre expérience sur la détection des données personnelles dans les textes nous ont permis d'éprouver l'efficacité de

entité d'intérêt	précision	rappel	f-score
email	1.0000	1.0000	1.0000
function	0.9638	0.8088	0.8795
human	0.9238	0.9222	0.9230
location	0.9382	0.9361	0.9372
organization	0.7988	0.8966	0.8449
phone_number	1.0000	0.9927	0.9964
project	0.2837	0.4758	0.3554
reference_cedex	0.9776	0.8973	0.9357
reference_code_postal	0.9140	0.9798	0.9458
reference_cs	1.0000	0.0541	0.1026
reference_user	1.0000	0.8182	0.9000
social_network	1.0000	0.9286	0.9630
url	1.0000	1.0000	1.0000

TAB. 3 – *Résultats du run3, par type d'entité*

cette approche. Notre équipe est arrivée première avec les meilleures performances, à l'aide d'une approche semblable à celle de la seconde équipe du classement.

Nous avons fait le choix de décomposer notre système en deux parties. La première regroupe les modules de détection qui constitue une réponse plus proche des attentes globales du sujet du défi : "la reconnaissance d'entités d'intérêts dans les signatures d'e-mails dans le but de structurer l'information et de la stocker en base de données"(Cousot et al., 2022) . En condition réelle, la tâche nécessite à la fois de détecter les mentions qui sont des entités de celles qui ne le sont et de les catégoriser. Ce double objectif est adressé par la première partie de notre système qui identifie les mentions candidates en plein texte puis les catégorise. Cela fait également écho à nos problématiques en tant qu'industriel. Notre approche a associé des règles, des CRF et des Transformers qui ont donné des résultats satisfaisants sur les différents types d'entité sauf pour les *Project*, et des résultats améliorables pour les *Organization*, *Function* et *Location*. La seconde partie de notre système s'attache uniquement à la tâche de classification puisqu'elles s'appuient sur les entités directement identifiées dans le JDA. Les règles créées ont porté majoritairement sur les entités qui ont présenté des résultats à perfectionner en sortie des modules de détection.

Par ailleurs, nous avons rencontré un certain nombre de nouveaux cas dans les données comme les caractères accentués dans les adresses e-mail ou imaginé de nouvelles règles comme la détection des tokens précédents pour renforcer la détection des *Location*. Ce défi a donc été pour nous l'occasion de découvrir de nouvelles pistes d'amélioration pour nos propres travaux.

Remerciements Nous tenons à remercier les organisateurs du Défi TextMine'23 pour avoir organisé ce challenge et pour les efforts fournis lors de la préparation des différents jeux de données mis à notre disposition.

Références

Bendahman, N., K. Cousot, et C. Lopez (2022). Reconnaissance d'entités d'intérêt dans les signatures d'e-mails à partir d'un jeu de données authentique. In *TextMine'22*.

Cousot, K., C. Lopez, et P. Cuxac (2022). Défi textmine'23 - reconnaissance d'entités d'intérêts dans les signatures d'e-mails. In *TextMine'23*.

Derczynski, L., K. Bontcheva, et I. Roberts (2016). Broad Twitter corpus : A diverse named entity recognition resource. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics : Technical Papers*, Osaka, Japan, pp. 1169–1179. The COLING 2016 Organizing Committee.

Devlin, J., M. Chang, K. Lee, et K. Toutanova (2018). BERT : pre-training of deep bidirectional transformers for language understanding. *CoRR abs/1810.04805*.

Guo, Y., R. J. Gaizauskas, I. Roberts, G. Demetriou, et M. Hepple (2006). Identifying personal health information using support vector machines.

Honnibal, M., I. Montani, S. Van Landeghem, et A. Boyd (2020). spacy : Industrial-strength natural language processing in python.

Jia, C., X. Liang, et Y. Zhang (2019). Cross-domain NER using cross-domain language modeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy, pp. 2464–2474. Association for Computational Linguistics.

Lafferty, J. D., A. McCallum, et F. C. N. Pereira (2001). Conditional random fields : Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, San Francisco, CA, USA, pp. 282–289. Morgan Kaufmann Publishers Inc.

Lawson, N., K. Eustice, M. Perkowitz, et M. Yetisgen-Yildiz (2010). Annotating large email datasets for named entity recognition with Mechanical Turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, Los Angeles, pp. 71–79. Association for Computational Linguistics.

Li, J., A. Sun, J. Han, et C. Li (2018). A survey on deep learning for named entity recognition. *CoRR abs/1812.09449*.

Liu, Z., B. Tang, X. Wang, et Q. Chen (2017). De-identification of clinical notes via recurrent neural network and conditional random field. *Journal of biomedical informatics 75S*, S34—S42.

Martin, L., B. Muller, P. J. Ortiz Suárez, Y. Dupont, L. Romary, É. de la Clergerie, D. Seddah, et B. Sagot (2020). CamemBERT : a tasty French language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online, pp. 7203–7219. Association for Computational Linguistics.

McDonald, D. (1993). Internal and external evidence in the identification and semantic categorization of proper names. In *Acquisition of Lexical Knowledge from Text*.

Segura-Bedmar, I., P. Martínez, et M. Herrero-Zazo (2013). SemEval-2013 task 9 : Extraction of drug-drug interactions from biomedical texts (DDIExtraction 2013). In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2 : Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, Atlanta, Georgia,

USA, pp. 341–350. Association for Computational Linguistics.

Shengyu, L., T. Buzhou, C. Qingcai, et W. Xiaolong (2015). Effects of semantic features on machine learning-based drug name recognition systems : Word embeddings vs. manually constructed dictionaries. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2 : Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pp. 848–865. Information 2015.

Sweeney, L. (1996). Replacing personally-identifying information in medical records, the scrub system. *Proc AMIA Annu Fall Symp. abs/1812.09449*, 333–7.

Tjong Kim Sang, E. F. et F. De Meulder (2003). Introduction to the CoNLL-2003 shared task : Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pp. 142–147.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, et I. Polosukhin (2017). Attention is all you need. *CoRR abs/1706.03762*.

Voita, E., D. Talbot, F. Moiseev, R. Sennrich, et I. Titov (2019). Analyzing multi-head self-attention : Specialized heads do the heavy lifting, the rest can be pruned. *CoRR abs/1905.09418*.

Summary

In this paper, we present the systems developed by the Nextino NLP team for the TextMine'23 challenge, a shared task on entities classification in e-mail signatures. Our solution relies on a hybridation approach which combines the strengths of several methods –regular expressions, CRF and Transformers– to fit the specificities of each type of entity. In order to improve the challenging entities, a post-processing step is performed. Our participation aims to address two issues: the classification of the entities, as part of the challenge and their detection from the raw text, as an underlying problem. We submitted 3 runs and ranked 1st on the leaderboard with a F-score of 0.908.

Index

Adam,Olivier, 14
Ayoub, Oussama, 41

Berkenbaum, Lara, 14
Bothua, Meryl, 51
Brassier, Maëlle, 74

Coch, José, 14
Cousot, Kevin, 3, 45
Cuxac, Pascal, 45

Fouqué, Nicolas, 70

Gernot, Tanguy, 60
Giguet, Emmanuel, 60
Goudjo, Asceline, 74
Guille, Adrien, 1

Hassani, Leila, 51
Hau Nguyen, Duc, 70

Klötzer, Victor, 70

Latour, Maryline, 32
Lemaire, Vincent, 45
Li, Ludovic, 41
Lopez, Cédric, 45

Mirzapour, Mehdi, 3
Moncuquet, Paul, 32
Moradi-Farisar, Nilofar, 3

Ragheb, Waleed, 3
Rodrigues, Christophe, 41

Suignard, Philippe, 51

Thomas, Hugo, 70
Travers, Nicolas, 41