



HAL
open science

Iterative and Semi-Supervised Design of Chatbots Using Interactive Clustering

Erwan Schild, Gautier Durantin, Jean-Charles Lamirel, Florian Miconi

► To cite this version:

Erwan Schild, Gautier Durantin, Jean-Charles Lamirel, Florian Miconi. Iterative and Semi-Supervised Design of Chatbots Using Interactive Clustering. *International Journal of Data Warehousing and Mining (IJDWM)*, 2022, 18 (2), pp.1-19. 10.4018/IJDWM.298007 . hal-03648041v2

HAL Id: hal-03648041

<https://hal.univ-lorraine.fr/hal-03648041v2>

Submitted on 27 Oct 2022

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

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



Distributed under a Creative Commons Attribution 4.0 International License

Iterative and Semi-Supervised Design of Chatbots using Interactive Clustering

Erwan SCHILD

erwan.schild@e-i.com

EURO INFORMATION DÉVELOPPEMENTS (COGNITIVE FACTORY)

Gautier DURANTIN

gautier.durantin@e-i.com

EURO INFORMATION DÉVELOPPEMENTS (COGNITIVE FACTORY)

Jean-Charles LAMIREL

jean-charles.lamirel@loria.fr

LORIA (SYNALP)

Florian MICONI (florian.miconi@e-i.com)

EURO INFORMATION DÉVELOPPEMENTS (COGNITIVE FACTORY)

DOI: 10.4018/IJDWM.298007

ABSTRACT

Chatbots represent a promising tool to automate the processing of requests in a business context. However, despite major progress in natural language processing technologies, constructing a dataset deemed relevant by business experts is a manual, iterative and error-prone process. To assist these experts during modelling and labelling, the authors propose an active learning methodology coined Interactive Clustering. It relies on interactions between computer-guided segmentation of data in intents, and response-driven human annotations imposing constraints on clusters to improve relevance. This article applies Interactive Clustering on a realistic dataset, and measures the optimal settings required for relevant segmentation in a minimal number of annotations. The usability of the method is discussed in terms of computation time, and the achieved compromise between business relevance and classification performance during training. In this context, Interactive Clustering appears as a suitable methodology combining human and computer initiatives to develop efficiently a useable chatbot.

Keywords: Active Learning, Constrained Clustering, Annotation, Business Relevant Dataset, Natural Language Processing, Intent Modelling, Business Expert

KEYWORDS

Active Learning, Annotation, Business Expert, Business Relevant Dataset, Constrained Clustering, Intent Modelling, Natural Language Processing

This article published as an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>) which permits unrestricted use, distribution, and production in any medium, provided the author of the original work and original publication source are properly credited.

INTRODUCTION

Conversational assistants, also called chatbots, offer a flexible medium of communication to access information using natural language. By providing an automated answer to common requests, they contribute to increased rapidity and availability of customer care services. They also guarantee a uniform and efficient treatment of simple requests. Therefore, the use of chatbot in industry has gained momentum over the last few years, for basic question answering, automation of customer requests or access to complex databases (Goasduff, 2019; Costello, 2019).

The growing importance of chatbots has been supported by a rapid development of Natural Language Processing algorithms over the last decade, enabling efficient classification of user requests (using Natural Language Classification) or retrieval of relevant information from sentences (using Named Entities Recognition). As the frameworks for classifying or extracting information grow stronger, the development of chatbots in industry has consensually focused on defining a dialog behaviour with the use of symbolic reasoning, in which user requests will be treated according to the definition of all request categories (coined *intents*) and named entities (e.g. mails, numbers, products, etc...) (Hoyt et al., 2016; Bocklisch et al., 2017; Alexa Internet, 2018). For instance, the request “*Can you play some jazz?*” can be modelled with one intent (*play music*) and one named entity (*jazz*). In practice, the set of intents is finite and relates to a specific business area (travel booking, banking assistance, etc.). This approach allows fast technical implementation and a good level of control over responses, which could explain its popularity.

Intent detection is generally implemented using supervised classification techniques, where classes represent intents (Adamopoulou & Moussiades, 2020). A dialog management system (e.g. a decision tree) then exploits the intents and named entities extracted from user requests to define the behaviour of the chatbot. Thus, at every step during design, a dataset labelled with intents and named entities is required to generate and maintain the classification model forming the core of the chatbot. Updating and annotating a dataset is usually considered an iterative process involving steps described by the acronym MATTER (Model, Annotate, Train, Test, Evaluate, and Revise) (see Stubbs (2013)). Following this approach, phases of dataset design and annotation are most often the result of manual work, and this methodology has several limitations:

- Prior to starting annotation, a model grouping data in relevant intents has to be defined: when performed manually, this task relies exclusively on the knowledge of the types of requests detained by one or several business experts;
- Once the modelling of intents is achieved, the task of labelling data requires thorough understanding of this model to avoid errors: when the knowledge of the intents model is not well integrated by one of the annotators, or uniformly shared between them, there is a risk of introducing intra-individual or inter-individual inconsistencies in the dataset;
- During initialization or maintenance of the model, the scope of the chatbot may change: in this case, any change to the intents model implies an additional cost of relabelling the dataset accordingly.

The approach hitherto described is therefore time consuming, expensive, dependent of human judgment and has low robustness to changes.

To provide assistance during this annotation task, one possible solution is to introduce computer initiatives. This can be implemented using unsupervised classification (*clustering*) with automatic partitioning of data based on their intrinsic similarities. However, the similarities exploited by unsupervised text classification algorithms are usually either lexical or syntactical, and do not guarantee that the data belong to a similar business domain. Consequently, the produced results are often qualified as irrelevant by business experts.

To overcome this limitation and increase business relevance, there are promising approaches based on constrained clustering to influence data partitioning with human knowledge. As an example, (Lampert et al., 2019) propose a collaborative clustering approach integrating user constraints to improve performance of a topographical clustering task from satellite images. However, such constrained annotation methods are harder to apply to natural language: data cannot be observed all at once by the human when defining constraints, as can be done with an image.

In Schild et al. (2021a, 2021b), the authors proposed an active learning method adapted from images constrained clustering to assist questions annotation task. This approach, coined *Interactive Clustering*, is composed of several iterations involving data sampling heuristics to guide expert annotations and a constrained clustering algorithm to partition data with expert corrections. The study showed that adequate selection of implementation parameters could lead to a relevant iterative definition of intents on a small dataset. While the article defines a methodology that could be suited to define intents set iteratively, this conclusion has to be confirmed on a larger dataset containing more data and classes. The usability of this method in a realistic setting was also not explored in this study, for instance in terms of time consumption or classification performance of the resulting intents model.

In this article, the authors propose an extension of the previous study to investigate the suitability of *Interactive Clustering* for the iterative design of conversational agents. Stemming from practical organization constraints encountered when developing chatbots in a business context (see Finlayson & Erjavec (2016)), the methodology is applied on a realistic French dataset representative of requests in the banking domain (including ground truth labels proposed by domain experts). In particular, the ability of the method to take advantage of both computer guidance and human expertise is tested by measuring how iterations of *Interactive Clustering* converge to the ground truth over time. The authors also propose to verify the usability of *Interactive Clustering* by analysing the influence of implementation parameters (i.e. algorithms chosen to implement the method) on its convergence and computation times, as well as on the ability to provide a segmentation of data enabling high classification performance, while maintaining business relevance.

BACKGROUND

How to Design a Chatbot in Practice?

The design of a chatbot as a conversational tool to facilitate operations in a given business area relies on (1) the definition of a desired user experience; (2) the modelling of the business knowledge relative to this experience; and (3) the implementation of a technical platform supporting the dialogic experience. Constructing these solutions along these three constraints requires interactions between end users, business experts and data scientists. In practice, and from the authors' experience, the design follows the steps below (see also Finlayson & Erjavec (2016) and Stubbs (2013)).

Step 1: Defining the business area.

A chatbot is more efficient on a narrow business scope, where the complexity of the modelling of relevant knowledge will be limited. This step involves business experts who can validate the relevance of information considered for the chatbot.

Step 2: Collecting a dataset of questions related to this area.

Questions can come from different sources: user surveys, experts' insights, database extractions, or web scrapping. Data collection should include a filtering step to exclude out of scope data (e.g. questions not related to the specified area).

Step 3: Modelling the set of intents from the dataset.

Intents can be defined as the action or information expected by the user when formulating a request. Business experts define the set of intents according to the data collected and their domain knowledge. A

chatbot dedicated to requests related to mortgage loans may for example include intents like “*loan subscription*”, “*contract negotiation*”, or “*customer state consultation*”.

Modelling intents can be complex, because of the elasticity of business knowledge: in most cases, experts will become aware of certain requests and incorporate them in the model only when presented with actual questions. The types and number of requests that need to be integrated therefore varies greatly as experts explore the dataset and interact with other experts. From the authors’ experience, depending on the level of detail desired during modelling, experts can define up to 100 intents from a 10.000 questions dataset.

Step 4: Labelling the dataset.

Labelling the dataset is a necessary step towards training the intents model. During annotation, business experts must maintain in memory a representation of the entire set of intents. The complexity of the model, in addition to the task being time consuming and repetitive, makes annotation an error-prone task, that can be subjective and introduce inconsistencies.

In practice, the annotation step questions the modelling of intents, and experts may decide to revise the set and the definition of intents (thus rolling back to step 3).

Step 5: Training and implementing the chatbot.

Once labelling is complete, a supervised classification model is trained and the dialog behaviour is defined from the intents (e.g. using a dialog decision tree). At this stage, training the model often yields poor performance due to the complexity of the intents model or inconsistencies in annotation.

In practice, this step also challenges the intents modelling, and the design process can roll back to steps 3 and 4 several times before it reaches acceptable performance.

Step 6: Perform continuous improvement and intents scope maintenance.

Once the chatbot is in production, performance is monitored to define areas of improvement (increase performance, reduce dialog ambiguities). In addition, experts have to follow the evolution of business area in order to handle new requests with intent creation or modification.

Computer-assisted modelling using Unsupervised Learning

To assist humans during annotation, one option is the introduction of machine initiatives. Using unsupervised classification, an algorithm clusters data according to their intrinsic similarities and can suggest efficient intents grouping. Several known algorithms and methods can be used:

- **K-Means Clustering** (MacQueen, 1967): A common clustering method that relies on minimizing intra-class inertia by assigning each data to the nearest cluster barycentre. This algorithm is one of the most commons because of its simplicity and computation speed.
- **Hierarchical Clustering** (Murtagh & Contreras, 2012): An iterative method of merging most similar data into clusters. Several types of similarity links can define the merging strategy. (Single link: merge the two clusters with the closest borders. Full link: merge the two clusters with the closest opposite borders. Medium link: merge the two clusters with the closest cluster centers. Ward link: merge the two clusters that will result in the most compact cluster.).
- **Spectral Clustering** (Ng et al., 2002): A method that relies on modelling the similarity matrix between data by its eigenvectors and grouping them with a K-Means type algorithm. This approach can handle clusters with complex shapes.

However, despite the guaranteed adequate performance during classification, suggestions of clustering algorithms can be unreliable. Known limitations include difficulty to handle noisy or high-dimensional data, and their inability to exploit metrics suited to the problem they are dealing with. In addition, one of

their major drawbacks lies in the lack of business relevance of the suggestions, because clustering algorithms cannot extract specific knowledge on the business area without human intervention (Xu & Tian, 2015). All of these problems are often important in natural language processing tasks, which reduces the usefulness of clustering methods in this field.

Constrained clustering is a semi-supervised variant of clustering methods. This alternative consists in influencing clustering by specifying a priori links between observations called “*constraints*”. Expert annotations or heuristics can impose two types of constraints (Wagstaff & Cardie, 2000): “*MUST-LINK*”, i.e. data are similar, and “*CANNOT-LINK*”, i.e. data are not similar. These constraints can efficiently influence clustering results, and experts can thus introduce nuances that the algorithm would not have detected on its own.

The previously cited examples of clustering all have a constrained equivalent:

- **COP K-Means Clustering** (Wagstaff et al., 2000): During assignment of data to the nearest cluster, constraints are checked to correct the assignment when needed. The implementation of this version is simple, but its execution can lead to unresolved contradictions (when all clusters have at least one conflict with the current data to affect). An adaptation is to create an additional cluster to collect the conflicting data.
- **Constrained Hierarchical Clustering** (Davidson & Ravi, 2005): The strategy is to merge the clusters with “*MUST-LINK*” constraints first and to prevent the merge if two clusters have a “*CANNOT-LINK*” constraint. To implement it, a simple way is to adapt the calculation of similarity score between the clusters.
- **Constrained Spectral Clustering** (Kamvar et al., 2003): To handle constraints, the value of the similarity matrix is forced to zero (respectively one) if the two observations are linked by a “*MUST-LINK*” constraint (respectively “*CANNOT-LINK*”). The adaptation therefore requires little effort, but the addition of a constraint can lead in some cases to a radical change in algorithm results.
- Other examples are described in Lampert et al. (2018).

These algorithms require constraints annotation, a binary classification task that does not require abstract data modelling. During constraint annotation, the attention of the expert remains directed towards real-life examples taken from the categories rather than their abstract representations. For instance, in the project working on satellite images clustering presented by Lampert et al. (2019), the attention of the expert is dedicated to linking actual image zones together, rather than manipulating abstract representations of these topographical zones (forest, water, etc...). However, imposing adequate constraints is more complex in natural language processing, as observations must be handled individually (whereas images can be processed as a whole). As a result, the definition of adequate constraints requires rather an iterative process, and to define how individual data points will be presented to the user.

Computer-assisted modelling using Active Learning

Active learning relies on coordinating human knowledge and machine capabilities. In this iterative process, the human adjusts the result proposed by the machine, and the machine then uses these corrections to improve its subsequent iterations (Settles, 2010). To optimize the relevance of the corrections made by humans, the active learning process typically uses an oracle responsible for the improvement strategy. It aims to achieve maximum efficiency during human-machine interactions.

The oracle can exploit different strategies to select relevant data, such as verifying the confidence level of the predictions, estimating error correction on results or maximizing exploration of the corpus (Settles, 2010). By following these strategies, the annotation is no longer limited to a manual or random process, but it uses improvement heuristics that require humans to introduce knowledge.

PRINCIPLES OF INTERACTIVE CLUSTERING

Combining human expertise and computer suggestions to improve clustering

The purpose of this article is to define and experiment an alternative to manual annotation of intents for text data. To that end, the authors propose to require from annotators only inputs that are relevant to their knowledge domain. In practice, this means that instead of having business experts define upfront a complex data modelling that would require to anticipate technical constraints (such as the capacity to fit a model to their data), the proposed methodology focuses on their ability to discriminate requests based on the response expected from a business perspective.

In accordance with this principle, the authors propose a novel iterative active learning methodology coined "*Interactive Clustering*", involving three sub-steps:

1. **Constraints Sampling:** A heuristic suggests pairs of questions that the expert has to discriminate. Heuristics can be based on different strategies, and the authors propose to consider the following:
 - a. *Random:* A basic strategy that randomly selects pairs of questions;
 - b. *Random in Same Cluster:* A strategy aimed at verifying the homogeneity of clusters by randomly selecting pairs of data from a same cluster;
 - c. *Closest Neighbors in Different Clusters:* A strategy aimed at validating the position of the borders between clusters by selecting the closest data from two different clusters;
 - d. *Farthest Neighbors in Same Cluster:* A strategy aimed at ensuring that a cluster does not absorb clusters at its borders by selecting the most distant pairs of data from the same cluster.
2. **Constraints Annotation:** The expert discriminates each pair of selected questions according to a business characteristic. For instance, the authors propose to discriminate couples of questions based on the response they would require. In this context, a "*MUST LINK*" constraint is placed each time the requests require identical actions (and "*CANNOT LINK*" otherwise). Constraints' transitivity is used, i.e. the manager makes deductions on the annotated constraints. For example, if d_1 and d_2 have a "*MUST-LINK*" constraint, and d_2 and d_3 have a "*CANNOT-LINK*" constraint, then d_1 and d_3 will have a "*CANNOT-LINK*" constraint applied by transitivity.
3. **Constrained Clustering:** The computer uses all annotated constraints to improve iteratively the relevance of clustering results. Several constrained algorithms are proposed:
 - a. COP K-Means clustering;
 - b. Hierarchical clustering;
 - c. SPEC Spectral clustering.

The process is initialized with an unconstrained clustering to provide a first (and likely little relevant) data partitioning. During each iteration of *Interactive Clustering*, the computer selects a pair of questions using a predetermined heuristic, the expert discriminates them by annotating binary constraints, and the clustering uses the constraints to perform a corrected data partitioning. Thus, the relevance of clustering results should increase over iterations.

Using this methodology, the annotator does not need to manage a dataset and requires little knowledge in data science: they simply have to express their knowledge of the relevant business domain. Furthermore, a preliminary definition of possible intents is no longer necessary to start annotating: the intent structure will be determined over iterations through clustering.

The authors propose a *Python* implementation (Van Rossum & Drake, 2009) of this methodology available in Schild (2021d). The library is composed of three modules, one per sub-step of the *Interactive Clustering* methodology.

How to Design a Chatbot using Interactive Clustering in Practice?

The steps below are adapted from the standard organization of the chatbot design process, to integrate *Interactive Clustering*.

Step 1: Define the specific business area.

See above, no changes.

Step 2: Collect a dataset of questions related to this area.

See above, no changes.

Step 3: Perform iterative and interactive annotation with *Interactive Clustering*.

The annotator runs several iterations of constraints sampling (1), constraints annotation (2) and constrained clustering (3). Over iterations, the business relevance of the clustering should improve by incorporating more constraints from the expert. This task does not use any predefined intent modelling and uses binary constraints based on response similarity.

Step 4: Perform statistical and semantic validation.

This step focuses on validating the relevance of results obtained after a few iterations of *Interactive Clustering*.

The simplest check is a statistical validation of the data partitioning a *k-fold* cross validation checks the statistical consistency. Poor performance can be a sign that too many or inconsistent constraints have been placed on the data.

Then, another check is necessary to confirm the clusters business relevance: This analysis cannot be automated because it requires clusters inspection (lexical fields' analysis, relevant patterns detection, semantic consistency analysis, etc.). If the analysis is not satisfying, there are probably not enough annotated constraints for the *Interactive Clustering* to converge.

Step 5: Train and implement the chatbot with the labelled dataset.

After annotating with *Interactive Clustering*, clusters are named according to the discovered intents, and a supervised classification model is trained with these intents. Using this model, the dialog behaviour of the chatbot can be defined. This task is made easier since this method tends to group together questions that yield similar responses.

Step 6: Perform continuous improvement and intents scope maintenance.

See above, no changes.

CONVERGENCE AND IMPLEMENTATION TESTS

Hypotheses

This article aims to study the following two hypotheses:

Hypothesis One: An annotation methodology based on *Interactive Clustering* implementation can converge to a business relevant ground truth.

Hypothesis Two: The convergence speed of *Interactive Clustering* methodology depends on several implementation parameters. The authors specifically study the influence of data *preprocessing*, data *vectorization*, constraints *sampling* strategy, and constrained *clustering* algorithm.

Methods

To test the article hypotheses, the authors propose an experiment of chatbot dataset annotation. This experiment consists in performing *Interactive Clustering* iterations in order to annotate an unlabelled dataset, starting from no known constraints and ending when all the possible constraints between questions are defined.

The human annotator is simulated by the algorithm, and annotations are made by comparing with ground truth labels: two questions are annotated with a “*MUST-LINK*” if they come from the same intent, and with a “*CANNOT-LINK*” constraint otherwise. With this automatic annotation, no conflict can occur.

In this article, the influence of the following parameters is studied:

- i. **Four Levels of Data Preprocessing:** (a) *no preprocessing*; (b) *simple preprocessing* (lowercasing, accent deletion, punctuation deletion, whitespace deletion); (c) *lemmatized preprocessing* (simple preprocessing and token lemmatization); and (d) *filtered preprocessing* (simple preprocessing and restriction to first-order token in the syntactic dependency tree).
 - **Implementation:** use of spacy lemmatizer and dependency parser (Honnibal & Montani, 2017).
- ii. **Two Levels of Data Vectorization:** (a) *tfidf* (vectors based on terms frequency) and (b) *fr-core-news-md* model (pre-trained spacy language model).
 - **NB:** A model of pre-trained vectors on a banking corpus can be used, but no French model is available for this experiment. Such a model could be considered in a later study.
 - **Implementation:** use of *Scikit-Learn* TFIDF vectoriser (Pedregosa et al., 2011) and *spaCy* French language model (Honnibal & Montani, 2017).
- iii. **Four Levels of Constraints Sampling:** (a) sampling of *random pairs*; (b) sampling of *random pairs from a same cluster*; (c) sampling of *closest pairs from different clusters*; and (d) sampling of *farthest pairs from a same cluster*.
 - **NB:** For this study, the annotation batch size is set to 50 pairs of questions.
- iv. **Six Levels of Constrained Clustering:** (a) *COP K-Means* clustering; (b-e) *Hierarchical clustering* (four similarity links: *single*, *complete*, *average* and *ward*); and (f) *SPEC-Spectral* clustering
 - **NB:** For this study, it is assumed the number of clusters is known and set the ground truth intents number.

Altogether, there were 192 possible combinations of parameters. Each configuration was repeated 5 times.

The relevance of data segmentation is measured using *homogeneity*, *completeness*, and *v-measure*, computed on the clustering results of each iteration. Measures are obtained through comparison with a ground truth, corresponding to annotations by business experts prior to the experiment (with no computer guidance). In this study, the following performance thresholds are considered:

- **Complete Annotation:** The number of iterations required to annotate all constraints. In this case, the annotator defines the link between every pair of questions in the dataset. The clustering therefore becomes a deterministic graph traversal problem.
- **Sufficient Annotation:** The number of iterations required to reach a *v-measure* of 100%, corresponding to complete agreement with the ground truth.
- **Partial Annotation:** The number of iterations required to reach a *v-measure* of 80%.

To analyse the convergence speed and the effect size of the implementation parameters on the number of annotations required, the authors perform repeated measures ANOVA in R (R Core Team, 2017). Post-hoc comparisons are performed using Tukey HSD procedure.

Finally, the optimal set of parameters according to statistical analysis is selected to train a candidate intents classifier. Intent classification is implemented from the vectors extracted during *Interactive Clustering*, using a 5-fold cross-validation and a SVM from the *Scikit-Learn* framework (Pedregosa et al., 2011).

The implementation of this experimental protocol is available in Schild (2021e, *in press*). Computations are parallelized on 24 CPU (Intel(R) Xeon(R) CPU E5-2660 v4 @ 2.00GHz), one worker per CPU.

Dataset description

The ground truth used for this experiment is available in Schild (2021c). It relies on a French dataset of 500 questions dealing with typical bankcard management requests. Prior to the experiment, the dataset was analysed and annotated manually by business experts. It was divided into 10 intents of 50 questions each. Sample questions from each intent are shown in Table 1.

To comply with the previously defined instructions on constraints annotation, intents were created by grouping questions requiring similar responses. Thus, a pair of questions coming from the same intent can be annotated by a “*MUST-LINK*” constraint (similar responses according to an expert), or a “*CANNOT-LINK*” constraint otherwise.

Table 1. Extracts from the French dataset of usual bank card requests.

Intent Name	Example	Example translation
<i>card loss or stolen</i>	« Comment signaler une perte de carte de paiement ? »	“How to report a loss of payment card?”
<i>card swallowed</i>	« Comment récupérer une carte avalée ? »	“How to retrieve a swallowed card?”
<i>card ordering</i>	« Je souhaite changer de carte bancaire. »	“I want to change my bank card.”
<i>bank balance consultation</i>	« Où retrouver le solde de mon compte ? »	“Where can I find my account balance?”
<i>card insurance ganrantee</i>	« Que couvre ma carte bancaire en cas d'hospitalisation ? »	“What does my bank card cover in the event of hospitalization?”
<i>card unlocking</i>	« Ma carte a été suspendue suite à un mauvais code, puis-je la réactiver ? »	“My card has been suspended due to a wrong code, can I reactivate it?”
<i>virtual card management</i>	« Comment faire pour créer une carte de paiements virtuelle ? »	“How to create a virtual payment card?”
<i>bank overdraft management</i>	« Est-ce que j'ai un découvert autorisé ? »	“Have I an authorized overdraft?”
<i>payment limits management</i>	« Le plafond de ma carte est trop bas, que faire ? »	“My card limit is too low, what should I do?”
<i>contactless mode management</i>	« Je veux désactiver le sans contact sur ma carte. »	“I want to deactivate contactless on my card.”

Experimental Results

All trials of the *Interactive Clustering* experiments converge towards ground truth:

- **Initialization Step:** With an unconstrained clustering, the average *v-measure* is 19.05% ($min = 03.42\%$, $max = 47.75\%$, $\sigma = 13.38\%$).

- **Partial Annotation:** To reach 80% of *v-measure*, experiments took on average 59.04 iterations ($min = 11, max = 315, \sigma = 42.14$), being 2951.81 annotations. The fastest experiment required 550 annotations, including 269 “MUST-LINK” constraints.
- **Sufficient Annotation:** To reach 100% of *v-measure*, experiments took on average 76.29 iterations ($min = 19, max = 328, \sigma = 46.44$), being 3801.19 annotations. The fastest experiment required 950 annotations, including 641 “MUST-LINK” constraints.
- **Complete Annotation:** To reach the annotation completeness, experiments took on average 88.98 iterations ($min = 20, max = 394, \sigma = 68.21$), being 4431.34 annotations. The fastest experiment required 1000 annotations, including 668 “MUST-LINK” constraints.

Table 2 describes the influences of parameters on the iterations number needed to reach 80% of agreement with the ground truth. Statistical analysis highlights the significant main effects on the partial annotation convergence of *preprocessing* parameter ($\eta^2 = 0.992, p\text{-value} < 10^{-3}$), of *vectorization* parameter ($\eta^2 = 0.998, p\text{-value} < 10^{-3}$), of *sampling* parameter ($\eta^2 = 0.999, p\text{-value} < 10^{-3}$), and of *clustering* parameter ($\eta^2 = 0.999, p\text{-value} < 10^{-3}$). Post-hoc analysis of these effects shows that the best average setting is made of simple preprocessing, TFIDF vectorization, sampling of closest pairs from different clusters, and average-link or single-link hierarchical constrained clustering. The average number of iterations needed for these settings is 13 ($\sigma = 2.11$), being 650 (± 105) annotations.

Table 2. ANOVA (with post-hoc) estimating the effect of implementation parameters on the number of iterations needed to reach 80% of *v-measure*. Stars indicate significance levels ($\alpha=0.05$).

Factor Description		Descriptive Statistics			Effect Size Statistics	
Factor	Levels	Mean	Rank	SE	η^2	<i>p</i> -value
<i>preprocessing</i>	<i>simple-prep</i>	55.95	(1)	0.33	0.992	$7.72 e^{-13}$ (***)
	<i>lemma-prep</i>	57.25	(2)			
	<i>no-prep</i>	57.59	(2)			
	<i>filter-prep</i>	65.36	(4)			
<i>vectorization</i>	<i>tfidf</i>	55.00	(1)	0.30	0.998	$1.56 e^{-06}$ (***)
	<i>fr-core-news-md</i>	63.08	(2)			
<i>sampling</i>	<i>closest-in-different</i>	29.27	(1)	0.33	0.999	$< 2 e^{-16}$ (***)
	<i>random-in-same</i>	44.93	(2)			
	<i>random</i>	61.07	(3)			
	<i>farthest-in-same</i>	101.88	(4)			
<i>clustering</i>	<i>hierarchical-average</i>	44.89	(1)	0.32	0.999	$< 2 e^{-16}$ (***)
	<i>hierarchical-single</i>	45.27	(1)			
	<i>k-means-cop</i>	46.55	(3)			
	<i>hierarchical-ward</i>	65.79	(4)			
	<i>hierarchical-complete</i>	66.90	(5)			
	<i>spectral-spec</i>	84.83	(6)			

Table 3 describes the influences of parameters on the iterations number needed to reach 100% of agreement with the ground truth. Statistical analysis highlights the significant main effects on the sufficient annotation convergence of *preprocessing* parameter ($\eta^2 = 0.987, p\text{-value} < 10^{-3}$), of *vectorization* parameter ($\eta^2 = 0.991, p\text{-value} < 10^{-3}$), of *sampling* parameter ($\eta^2 = 0.998, p\text{-value} < 10^{-3}$), and of *clustering* parameter ($\eta^2 = 0.997, p\text{-value} < 10^{-3}$). Post-hoc analysis of these effects shows that the best average setting is made of lemmatized preprocessing, TFDIF vectorization, sampling of closest pairs from different

clusters, and K-Means constrained clustering. The average number of iterations needed for these settings is 34.6 ($\sigma = 7.44$), being 1730 (± 372) annotations.

Table 3. ANOVA (with post-hoc) estimating the effect of implementation parameters on the number of iterations needed to reach 100% of v-measure. Stars indicate significance levels ($\alpha=0.05$).

Factor Description		Descriptive Statistics			Effect Size Statistics	
Factor	Levels	Mean	Rank	SE	η^2	p-value
preprocessing	lemma-prep	72.86	(1)	0.32	0.987	$1.17 e^{-13}$ (***)
	simple-prep	73.30	(2)			
	no-prep	75.24	(2)			
	filter-prep	83.77	(4)			
vectorization	tfidf	71.16	(1)	0.36	0.991	$9.30 e^{-07}$ (***)
	fr-core-news-md	81.43	(2)			
sampling	closest-in-different	50.29	(1)	0.39	0.998	$< 2 e^{-16}$ (***)
	random-in-same	56.38	(2)			
	random	71.95	(3)			
	farthest-in-same	126.55	(4)			
clustering	k-means-cop	62.23	(1)	0.42	0.997	$< 2 e^{-16}$ (***)
	hierarchical-average	65.13	(2)			
	hierarchical-single	75.44	(3)			
	hierarchical-ward	80.44	(4)			
	hierarchical-complete	81.46	(4)			
	spectral-spec	93.06	(6)			

Table 4 **Erreur ! Source du renvoi introuvable.**describes the influences of parameters on the iterations number needed to reach annotation completeness. Statistical analysis highlights the significant main effects on the complete annotation convergence of preprocessing parameter ($\eta^2 = 0.909$, $p\text{-value} < 10^{-3}$), of vectorization parameter ($\eta^2 = 0.985$, $p\text{-value} < 10^{-3}$), of sampling parameter ($\eta^2 = 0.997$, $p\text{-value} < 10^{-3}$), and of clustering parameter ($\eta^2 = 0.999$, $p\text{-value} < 10^{-3}$). Post-hoc analysis of these effects shows that the best average setting is made of lemmatized, TFIDF vectorization, sampling of random pairs from same clusters, and K-Means constrained clustering. The average number of iterations needed for these settings is 32.6 ($\sigma = 1.14$), being 1630 (± 57) annotations.

Table 4. ANOVA (with post-hoc) estimating the effect of implementation parameters on the number of iterations needed to reach annotation completeness. Stars indicate significance levels ($\alpha=0.05$).

Factor Description		Descriptive Statistics			Effect Size Statistics	
Factor	Levels	Mean	Rank	SE	η^2	p-value
preprocessing	lemma-prep	85.89	(1)	0.42	0.909	$1.10 e^{-08}$ (***)
	filter-prep	89.55	(2)			
	simple-prep	89.64	(2)			
	no-prep	90.81	(4)			
vectorization	tfidf	85.50	(1)	0.39	0.985	$2.53 e^{-06}$ (***)
	fr-core-news-md	92.46	(2)			
sampling	random-in-same	57.23	(1)	0.42	0.997	$< 2 e^{-16}$ (***)
	random	72.80	(2)			
	closest-in-different	98.38	(3)			
	farthest-in-same	127.50	(4)			
clustering	k-means-cop	64.99	(1)	0.39	0.999	$< 2 e^{-16}$ (***)
	hierarchical-average	78.54	(2)			

	<i>hierarchical-ward</i>	81.31	(3)			
	<i>hierarchical-complete</i>	82.49	(3)			
	<i>spectral-spec</i>	93.78	(5)			
	<i>hierarchical-single</i>	132.75	(6)			

Figure 1 illustrates a comparison of average *v-measure* evolution for average settings and for identified best settings to reach 80% of *v-measure*, 100% of *v-measure* and annotation completeness.

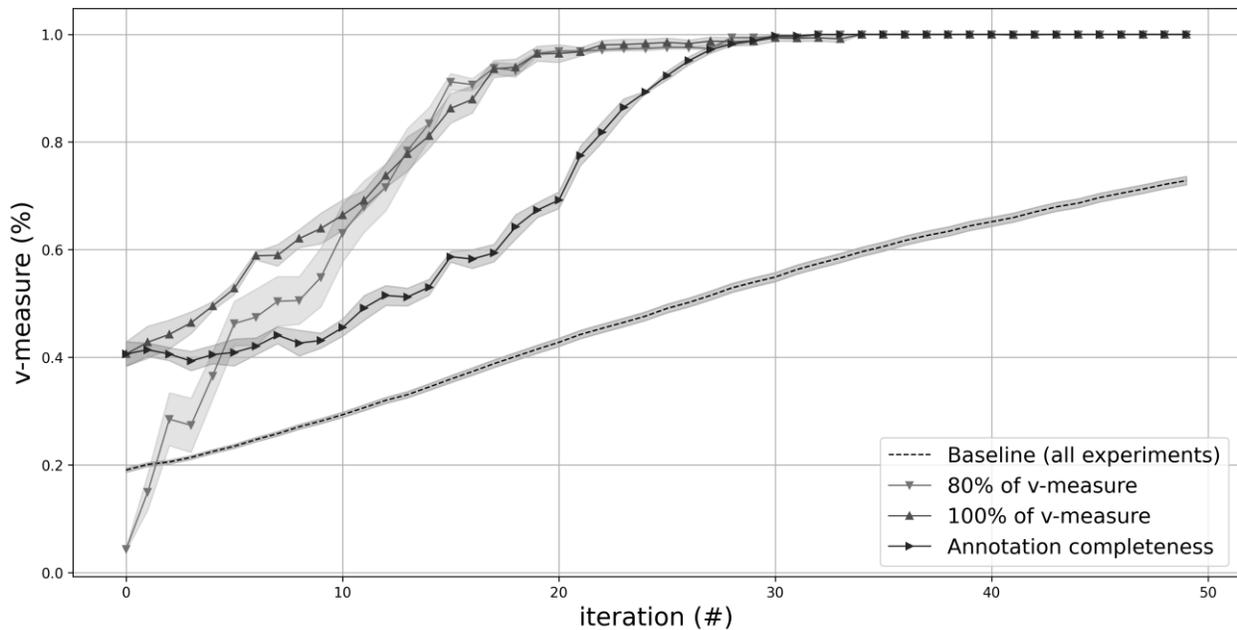


Figure 1. Average paths observed for optimal convergence of *v-measure* to a given objective (80%, 100%, or annotation completeness). Optimal parameters chosen for each objective correspond to highest-ranked ones in Tables 2, 3 and 4. The baseline corresponds to an average across all experiments, and error bars represent standard error of the mean.

Figure 2 illustrates a comparison of average clustering computation time evolution for the six examined clustering algorithms. Its shows that:

- Hierarchical clustering algorithms has a decreasing computation time according to the number of added constraints;
- K-Means clustering has an irregular but increasing computation time representative of assignation conflicts cases.

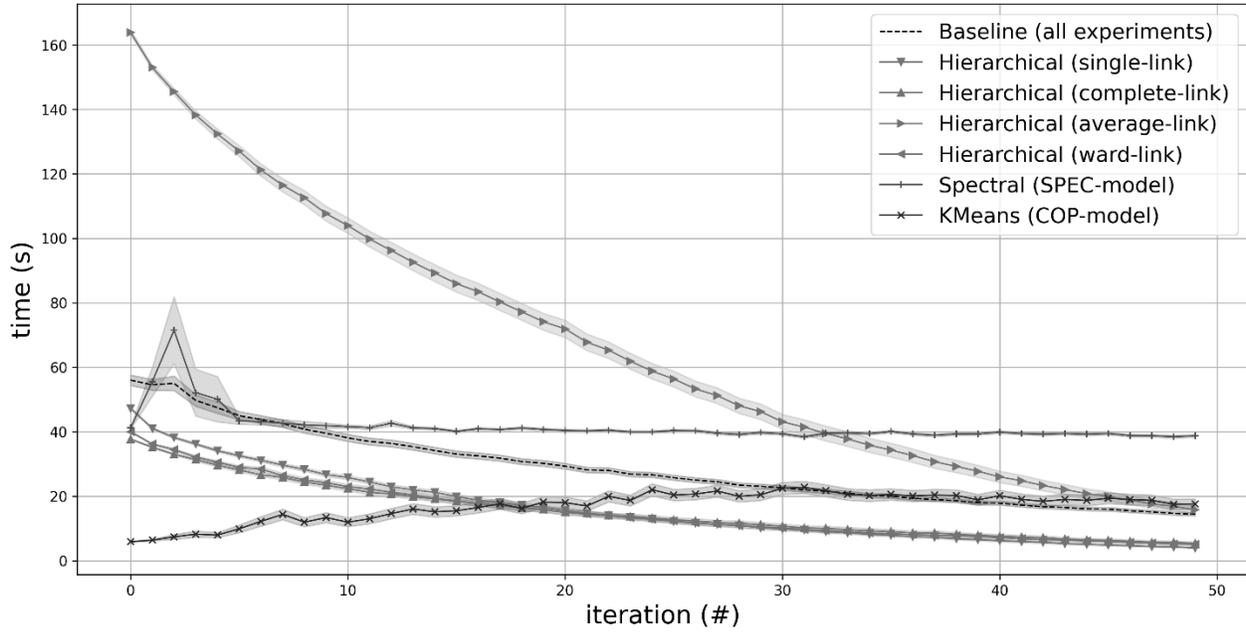


Figure 2. Evolution of computation time needed for adjusting the model across iterations of Interactive Clustering, depending on clustering algorithm. The baseline corresponds to an average across all experiments, and error bars represent standard error of the mean.

Figure 3 shows the results obtained from training a classifier on the dataset obtained using the optimal parameters set. Initial clustering of the dataset (with no input from the expert) yields a classification accuracy of 98%. This accuracy decreases as constraints are added by the expert (leading to higher business relevance of the segmentation). Analysis shows that a plateau of performance is achieved, with accuracy stabilizing around 94% starting from a v-measure of 0.75. Final accuracy obtained from the ground truth ($v\text{-measure} = 1$) is 94.2%.

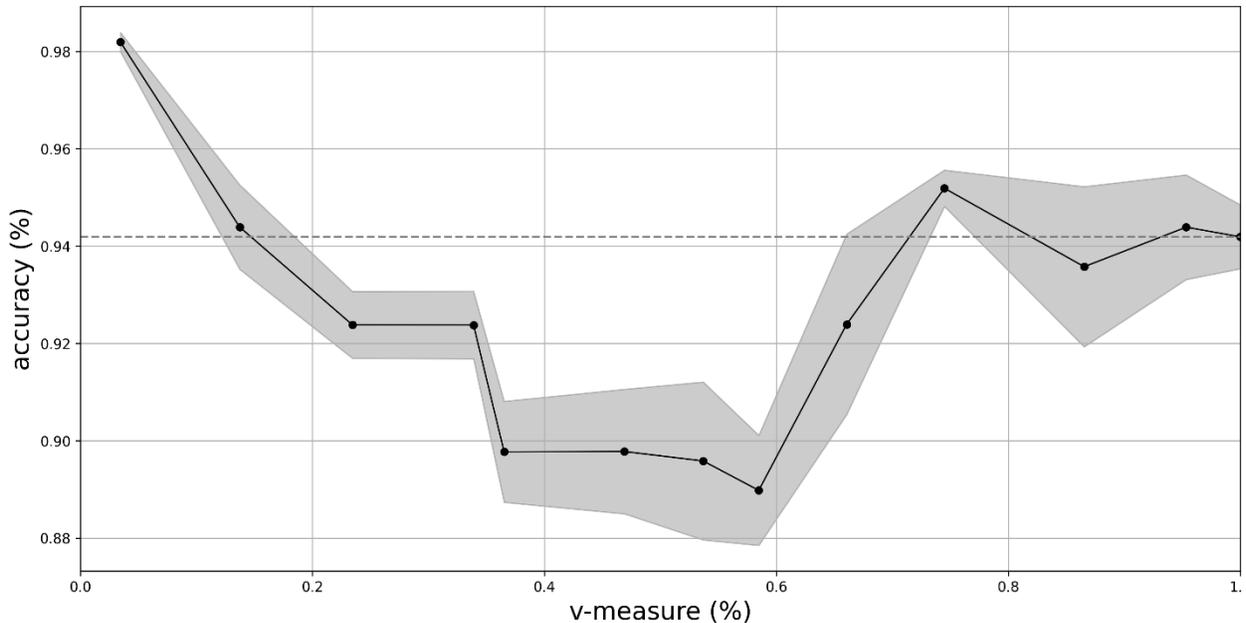


Figure 3. Evolution of accuracy obtained during training of an intent classification model, depending on the v -measure achieved during Interactive Clustering, for the optimal parameters set (simple preprocessing, TFIDF vectors, sampling of data along cluster borders and hierarchical single-link clustering). The curve starts from iteration zero (clustering without constraints). Error bars represent standard error of the mean. The dashed line represents performance achieved from Ground Truth.

Discussion

The aim of this article was to propose and experiment on a realistic dataset and methodology for the development of conversational agents in a business context. The method, adapted from Lampert et al. (2019) and coined *Interactive Clustering*, cycles through iterations composed of a constrained clustering phase (proposing a segmentation of unannotated requests samples), and a binary constraint annotation phase during which a business expert discriminates pairs of requests based on the expected response. Constraints take the form of “*MUST-LINK*” labels when the expected response is the same, and “*CANNOT-LINK*” otherwise.

Based on a realistic dataset of credit card-related questions, this article proposed to validate the capacity of the method to converge to a business-relevant segmentation of requests, and to test the influence of implementation parameters (preprocessing, vectorization, sampling heuristic, and clustering algorithm) on the convergence.

The results of the experiment confirm that regardless of the choice of implementation parameters, cycling through iterations of *Interactive Clustering* causes the data segmentation to converge to the human-made business-relevant segmentation used as a Ground Truth (see Figure 2). However, the shape and speed of convergence seems to depend on implementation parameters. A full dataset annotation is not possible because it requires too many constraints (4431 on average, 1630 at the optimum, see Table 4 for optimal settings). The authors therefore seek to perform a sufficient annotation (full subjective clustering correction) or a partial annotation (agreement between human annotation and clustering exploration).

As shown by the analysis of parameters proposed in Table 2 and Table 3, faster convergence can be obtained by using a simple (only character normalization) or lemmatized preprocessing, TFIDF vectors, and by annotating in priority pairs of data points selected at the border between clusters. These results are in agreement with optimal implementation parameters obtained on a smaller dataset in Schild (2021a, 2021b). COP K-Means and hierarchical clustering (single or average links) are the most adapted algorithms for faster convergence. However, even if COP K-Means faces some unstable iterations due to cluster assignment conflicts (cf. computation time variations), it is faster at first iterations, and is therefore more suited for quick integration of new constraints. On the other hand, hierarchical clustering algorithms are slower at first iterations, but have a decreasing computation time over iterations (see Figure 2).

As the methodology is based on mixed initiative from the human (to impose constraints) and the computer (to propose data segmentation), *Interactive Clustering* proposes an optimal compromise between business relevance of the segmentation and performance obtained during intent classification. In particular, analyses described in Figure 3 show that the algorithm starts from very high classification performance (but no business relevance) and progressively integrates user constraints to reach an adequate compromise with business relevance starting from 13 iterations (80% of v -measure).

These results suggest that *Interactive Clustering* can be an alternative to direct annotation during chatbot design. This method does not require prior modelling of intents by business experts (a task often complex and error-prone) and proposes to annotate based on the expected response to a request instead of requiring the business experts to memorize and share the same understanding of a complex pre-defined intent type system. In particular, the focus on response (rather than the content of the request) during annotation is in

line with the objectives of chatbot development and allows to distribute annotation tasks that correspond to the expertise of business users.

On this experiment on a realistic dataset of 500 questions, optimal segmentation with 80% of *v-measure* was reached after about 650 binary annotations, which is greater than the size of the dataset. However, the user experience associated with these annotations differs from direct annotation: while direct annotations require to place an abstract label chosen among a large set, *Interactive Clustering* constraints require simple discrimination between two sentences. By replacing complex categorization between multiple categories with discrimination between two cases, this approach therefore reduces the burden placed on the annotator's working memory. Consequently, the task requires less mental resources (Norman, 2013) and would facilitate the annotation process, in particular when the dataset and intent set are complex. The method would then seem particularly adapted as use cases grow complex, leading to greater difficulties to determine the intents set upfront.

Perspectives

While the *Interactive Clustering* method seems successful at providing a fluid user experience during initial definition of a chatbot on this use case, the authors have left for future investigation the study of the influence of potential annotation conflicts on convergence (for instance if conflicting constraints are defined). The provision of a cluster analysis tool could also be considered to help the task of labelling clusters.

To go further, although the authors recommend the use of optimal parameters described in Table 2 and Table 3, the exploration of other implementation parameters could potentially improve the convergence speed and shape. Their influence could also differ when applied to languages other than French.

Finally, several hypotheses concerning the annotator experience while performing interactive clustering still need to be verified in order to confirm the benefits of the proposed method. More particularly, future work can focus on estimating the time and mental load required to get a business relevant dataset using the proposed methodology. This inquiry would enable completion of the list of advantages, limits and scope of application of *Interactive Clustering* in a real-world setting.

CONCLUSION

This study was conducted in a context of growing automation in the treatment of customer requests, and the increasing efforts to develop conversational agents in a business operations context. It proposes a novel methodology of *Interactive Clustering* to support the efficient development of such conversational agents:

- It exploits computer suggestions from a constrained clustering algorithm to make simpler user experience during annotation by focusing on placing constraints the dataset rather than modelling the intents set upfront;
- It avoids error-prone situations occurring when the intents model is complex, by providing an evolving representation of intents: at each step of the process, a segmentation of the dataset is automatically kept up to date according to the constraints imposed by the expert, without requiring re-annotating the dataset;
- The experiment described in this article shows that the methodology can be tuned by choosing the optimal algorithms for text preprocessing, vectorization, clustering, and selection of constraints. The *Interactive Clustering* process then achieves a compromise between business relevance of the intents model and high potential for computer-based classification.

FUNDING AGENCY

Publisher has waived the Open Access publishing fee.

ACKNOWLEDGMENT

This research was supported by the French *Association Nationale de la Recherche et de la Technologie* (ANRT) [CIFRE n° 2019/0289]; by *Euro Information Développements*, fintech of *Crédit Mutuel Alliance Fédéral* banking group [CIFRE n° 2019/0289].

REFERENCES

- Adamopoulou, E., & Moussiades, L. (2020). An Overview of Chatbot Technology. In I. Maglogiannis, L. Iliadis, & E. Pimenidis (Eds.), *Artificial Intelligence Applications and Innovations* (pp. 373–383). Springer International Publishing. <https://doi.org/10/ghj8>
- Alexa Internet. (2018). *Keyword Research, Competitor Analysis, & Website Ranking | Alexa*. <https://www.alexacom>
- Bocklisch, T., Faulkner, J., Pawlowski, N., & Nichol, A. (2017). Rasa: Open Source Language Understanding and Dialogue Management. *ArXiv:1712.05181 [Cs]*. <http://arxiv.org/abs/1712.05181>
- Costello, K. (2019). Gartner Top Technologies and Trends Driving the Digital Workplace. *Gartner, Inc.* <https://www.gartner.com/smarterwithgartner/top-10-technologies-driving-the-digital-workplace/>
- Davidson, I., & Ravi, S. S. (2005). Agglomerative Hierarchical Clustering with Constraints: Theoretical and Empirical Results. *Springer, 3721*, 12. <https://doi.org/10/d4bsb4>
- Finlayson, M. A., & Erjavec, T. (2016). Overview of Annotation Creation: Processes & Tools. *ArXiv:1602.05753 [Cs]*. <http://arxiv.org/abs/1602.05753>
- Goasduff, L. (2019). Chatbots Will Appeal to Modern Workers. *Gartner, Inc.* <https://www.gartner.com/smarterwithgartner/chatbots-will-appeal-to-modern-workers/>
- Honnibal, M., & Montani, I. (2017). *spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing*.
- Hoyt, R. E., Snider, D., Thompson, C., & Mantravadi, S. (2016). IBM Watson Analytics: Automating Visualization, Descriptive, and Predictive Statistics. *JMIR Public Health Surveill*, 2(2). <https://doi.org/10.2196/publichealth.5810>
- Kamvar, S. D., Klein, D., & Manning, C. D. (2003). Spectral Learning. *Proceedings of the International Joint Conference on Artificial Intelligence*, 561–566.
- Lampert, T., Dao, T.-B.-H., Lafabregue, B., Serrette, N., Forestier, G., Crémilleux, B., Vrain, C., & Gançarski, P. (2018). Constrained distance based clustering for time-series: a comparative and experimental study. *Data Mining and Knowledge Discovery*, 32(6), 1663–1707. <https://doi.org/10/gfbpj8>
- Lampert, T., Lafabregue, B., & Gançarski, P. (2019). Constrained Distance based K-Means Clustering for Satellite Image Time-Series. *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, 2419–2422. <https://doi.org/10/ggx3tj>
- MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1(14), 281–297.
- Murtagh, F., & Contreras, P. (2012). Algorithms for hierarchical clustering: An overview. *Wiley Interdisciplinary Review: Data Mining and Knowledge Discovery*, 2, 86–97. <https://doi.org/10.1002/widm.53>
- Ng, A. Y., Jordan, M. I., & Weiss, Y. (2002). On Spectral Clustering: Analysis and an algorithm. In T. G. Dietterich, S. Becker, & Z. Ghahramani (Eds.), *Advances in Neural Information Processing Systems 14*

- (pp. 849–856). MIT Press. <http://papers.nips.cc/paper/2092-on-spectral-clustering-analysis-and-an-algorithm.pdf>
- Norman, G. (2013). Working memory and mental workload. *Advances in Health Sciences Education: Theory and Practice*, 18(2), 163-165.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.
- R Core Team. (2017). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. <https://www.R-project.org/>
- Schild, E., Durantin, G., Lamirel, J.-C., & Miconi, F. (2021a). *Conception itérative et semi-supervisée d'assistants conversationnels par regroupement interactif des questions*. RNTI E-37. <https://hal.inria.fr/hal-03133007>
- Schild, E., Durantin, G., & Lamirel, J.-C. (2021b). *Concevoir un assistant conversationnel de manière itérative et semi-supervisée avec le clustering interactif*. Atelier - Fouille de Textes - Text Mine 2021 - En conjonction avec EGC 2021. <https://hal.inria.fr/hal-03133060>
- Schild, E. (2021c). *French trainset for chatbots dealing with usual requests on bank cards*. <https://doi.org/10.5281/zenodo.4769949>
- Schild, E. (2021d). *cognitivefactory/interactive-clustering 0.4.2*. Zenodo. <https://doi.org/10.5281/zenodo.4775251>, <https://github.com/cognitivefactory/interactive-clustering>
- Schild, E. (2021e). *cognitivefactory/interactive-clustering-comparative-study 0.1.0*. Zenodo. <https://doi.org/10.5281/zenodo.5648256>, <https://github.com/cognitivefactory/interactive-clustering-comparative-study>.
- Settles, B. (2010). *Active Learning Literature Survey*. 67.
- Stubbs, A. C. (2013). *A Methodology for Using Professional Knowledge in Corpus* [Doctoral dissertation]. Brandeis University.
- Van Rossum, G., & Drake, F. L. (2009). *Python 3 Reference Manual* (CreateSpace).
- Wagstaff, K., & Cardie, C. (2000). Clustering with Instance-level Constraints. *Proceedings of the Seventeenth International Conference on Machine Learning*, 1103--1110.
- Wagstaff, K., Cardie, C., Rogers, S., & Schroedl, S. (2001). Constrained K-means Clustering with Background Knowledge. ICML.
- Xu, D., & Tian, Y. (2015). A Comprehensive Survey of Clustering Algorithms. *Annals of Data Science*, 2, 165–193.

Erwan Schild is completing his PhD in Machine Learning within a partnership between EURO INFORMATION (a subsidiary of Crédit Mutuel IT) and the LORIA (IT lab of Lorraine University). His PhD subject deals with semisupervised methods to assist textual annotation tasks in order to train chatbots. This PhD is funded by the French ANRT (“Association Nationale de la Recherche et de la Technologie”).

Gautier Durantin is the leader of the Cognitive Services team at Cognitive Factory in Strasbourg (France), developing NLP, image, and voice technologies for use in the Banking context. He obtained a Ph.D in Signal Processing and Neuroscience at the University of Toulouse, and is an alumnus of the Center of Excellence for the Dynamics of Language (Australia). His expertise covers Natural Language Processing, Ergonomics and Human-Computer Interaction.

Jean-Charles Lamirel is Lecturer in Computer Science with Research Accreditation at the University of Strasbourg as well as Sea-Sky Invited Professor at the University of Dalian (China). He is currently author of more than 160 research papers in International Conferences and Journals.

Florian Miconi has been involved in high energy physics research, he focused his PhD studies on the Higgs Boson using machine learning to explore Fermilab Tevatron’s data. He later joined Euro Information (Crédit Mutuel IT subsidiary) and is currently leading a unit composed of 3 teams specialized in cognitive technologies.