Contextualisation de Singularités en Temps-Réel par Extraction de Connaissances du Web des Données

Badre Belabbess[*,**] Jeremy Lhez [*]
Musab Bairat [**] Olivier Curé [**]

[*]Innovation Lab, ATOS, F-95870, Bezons, France
prénom.nom@atos.net,
[**]LIGM (UMR 8049), CNRS, F-77454, MLV, France.
prénom.nom@univ-paris-est.fr

Résumé. L'émergence de l'IoT et du traitement en temps-réel oblige les entreprises à considérer la détection d'anomalies comme un élément clé de leur activité. Afin de garantir une haute précision dans le processus de détection, des métadonnées fournissant un contexte spatio-temporel sur les mesures des capteurs sont nécessaires. Dans cet article, nous présentons un système générique qui aide à capturer, analyser, qualifier et stocker les informations contextuelles d'un domaine d'application donné. L'approche proposée est basée sur des méthodes sémantiques qui exploitent des ontologies pour évaluer la pertinence de l'information contextuelle. Après une description des composants principaux de l'architecture, la performance et la pertinence du système sont démontrées par une évaluation sur des ensembles de données du monde réel.

1 Introduction

Les capteurs de l'Internet des Objects (IoT) génèrent en continu de grandes quantités de données accumulées et traitées par des plates-formes spécialisées. L'analyse de ces données se fait par le biais de processus avancés basés sur de l'apprentissage automatique (*i.e.*, calcul numérique) ou des approches plus sémantiques (basées sur la représentation des connaissances et l'inférence). Parmi les problématiques phares, l'identification de singularités conduisant à la détection d'anomalies est un domaine de recherche d'actualité. En effet, ce sujet touche à des domaines aussi variés que la médecine (*e.g.*, identification de tumeurs malignes via imagerie IRM), la finance (*e.g.*, découverte de cas de fraudes lors de transactions financières), les technologies de l'information (*e.g.*, détection de piratage de réseaux informatiques).

Dans le cadre du projet Waves [1], nous nous sommes intéressés à la détection d'anomalies dans les grands réseaux d'eau potable gérés par un leader national expert dans le domaine de l'eau. La détection automatique de telles anomalies est une question importante à la fois sur le plan environnemental et économique. On notera que le volume de pertes d'eau potable enregistré dans le monde dépasse les 32 milliards de m3 / an (soit 14 milliards d'euros par an) dont 90 % reste difficilement identifiable en raison de la nature souterraine du réseau. Théoriquement,

1. https ://www.waves-rsp.org/

ces fuites d'eau peuvent être détectées en fonction de la pression et des mesures d'écoulement extraites des capteurs installés à des points stratégiques du réseau. Dans cet article, nous nous intéressons à un réseau national Français qui est constitué d'environ 100 000 km de canaux équipés de plus de 3 000 capteurs et distribuant de l'eau potable à plus de 12 millions de clients. Selon les experts, il est possible de garantir une grande précision lors du processus de détection si une contextualisation des mesures est effectuée lorsqu'une singularité apparaît. Par exemple, des signaux anormaux de haute pression ou de flux importants pourraient indiquer une fuite d'eau.

Cependant, dans de nombreux cas d'événements particuliers tels que les compétitions sportives, les rencontres culturelles, ou les catastrophes naturelles, ces singularités pourraient aisément s'expliquer rendant les réactions de l'exploitant du réseau plus efficaces. De plus, les conditions météorologiques telles que la canicule, un arrosage important ou un incendie d'origine criminelle impliquent l'utilisation de quantités importantes d'eau et ne sont donc pas de véritables anomalies. Par conséquent, une approche efficace de détection d'anomalies ne peut faire l'économie d'une contextualisation précise intégrant à la fois une dimension spatiale, une dimension temporelle et une dimension sémantique. Conçu pour être un système générique, Scouter vise à simplifier toutes ces tâches en proposant une implémentation efficace et en facilitant considérablement la configuration des composants.

2 Architecture

Scouter a été développé pour être un système complet qui peut traiter à la fois des événements statiques et dynamiques ainsi que les analyser à l'aide d'un puissant ensemble de fonctions du traitement du langage naturel (TLN) et de méthodes sémantiques avancées. Entièrement configurable, l'objectif principal de Scouter est d'extraire des données efficacement à partir de différentes sources dans le Web, les traiter rapidement afin de quantifier le potentiel de chaque événement à expliquer les anomalies détectées par la plate-forme. Les principaux composants de notre système sont les suivants : un ensemble de connecteurs de données Web, une unité d'analyse multimédia, une unité de géolocalisation, un centre de stockage, un gestionnaire de messages et un fournisseur de services Web.

Les connecteurs Web consomment les données provenant de différentes sources à une certaine fréquence et en fonction de configurations prédéfinies dans une interface Web. Ces sources incluent **(a) des réseaux sociaux** tels que Twitter et Facebook (*e.g.*, les citoyens commentant les fuites d'eau à proximité),**(b) des sources médiatiques** via des flux RSS de divers journaux (*e.g.*, un article du Monde mentionnant un incendie), **(c) des informations météorologiques** provenant d'API open-source (*e.g.*, les conditions climatiques lors d'un événement spécifique), **(d) des événements organisés** extraits de fournisseurs open-source (*e.g.*, des concerts, des expositions ou des événements sportifs), **(e) et des informations de profilage** extraites de DBpedia (*e.g.*, nombre d'habitants ou type de quartier).

Les concepts et propriétés utilisées pour rechercher des données sont représentés par une ontologie qui formalise les différentes relations d'appartenance, elle est détaillée dans la section 3. L'unité d'analyse médiatique synthétise les flux provenant de Kafka et s'appuie sur Apache Spark pour analyser les flux en temps réel. Ces derniers sont enregistrés comme des événements annotés d'une géolocalisation, d'une date de début et de fin ainsi que d'une description. Afin de filtrer les événements les plus pertinents sans conserver de doublons dans

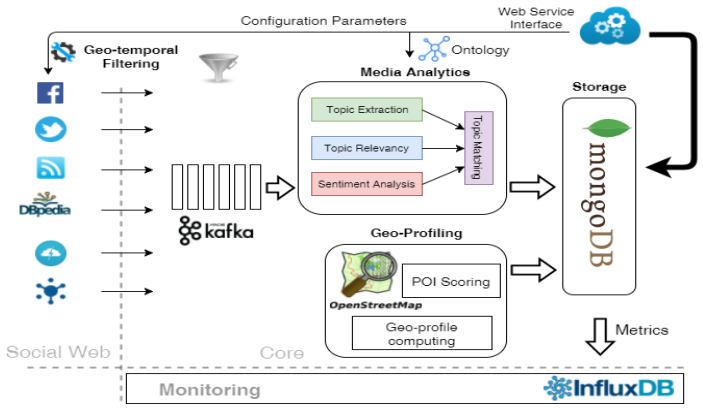

FIG. 1 – *Scouter Architecture*

la base de données, une approche d'extraction du résumé et une analyse de sentiment sont combinées. L'extraction de résumé analyse le texte des flux pour découvrir les occurrences de termes. Ensuite, le module de scoring tire parti des pondérations définies par l'utilisateur (*i.e.*, entre 0 et 1) associées aux concepts de l'ontologie pour fournir une note globale à chaque texte. L'analyse du sentiment classe les flux dans des catégories positives ou négatives en utilisant l'algorithme d'entropie maximum Adam Berger et Pietra (1996). Simultanément, l'unité de géo-profilage fournit des caractéristiques géographiques pour la zone analysée. Elle détermine le type de zone autour de l'emplacement de l'anomalie en générant un profil donné (*i.e.*, résidentiel, touristique, industriel ou agricole).

Suite aux étapes d'annotation et de scoring, les événements sont enregistrés dans une base de données distribuée orientée documents (MongoDB). Le résultat final obtenu est une contextualisation spatio-temporelle en temps réel pouvant expliquer une anomalie détectée dans le réseau d'eau potable. Scouter fournit également un outil de suivi des performances du système grâce à une panoplie de métriques telles que le temps d'exécution des requêtes ou la durée d'extraction des résumés. Ces métriques sont stockées dans une base de données orientée séries temporelles (InfluxDB) permettant un accès en lecture/écriture très rapide. Enfin, le composant de services Web est utilisé pour configurer le système de manière conviviale via une interface Rest.

3 Media Analytics & Traitement du Langage Naturel

Dans cette section, nous détaillons la méthodologie de collecte des données issues de différentes sources disponibles sur le Web.

3.1 Ontologie d'Extraction

Les systèmes de scrapping reposent généralement sur un fichier de configuration qui répertorie les propriétés des mots, des concepts ou des événements qu'il tentera d'extraire de S Sirisuriya (2015). Dans Scouter, l'extraction est optimisée et améliorée grâce à une ontologie

pré-construite qui énumère les principaux concepts que l'utilisateur recherche, elle permet d'organiser les différentes relations en deux dimensions :

Hiérarchie verticale : Un concept donné (*e.g.*, Feu) peut avoir plusieurs sous-concepts (*e.g.*, incendie, brasier, explosion) ou des alias et erreurs d'orthographe (*e.g.*, brazier, pheu).

Dépendance horizontale : Un concept peut avoir plusieurs propriétés qui décrivent un état spécifique durant une période données. Par exemple, l'eau peut être potable, mais peut également être en train de fuir ou avoir une couleur/odeur spécifique.

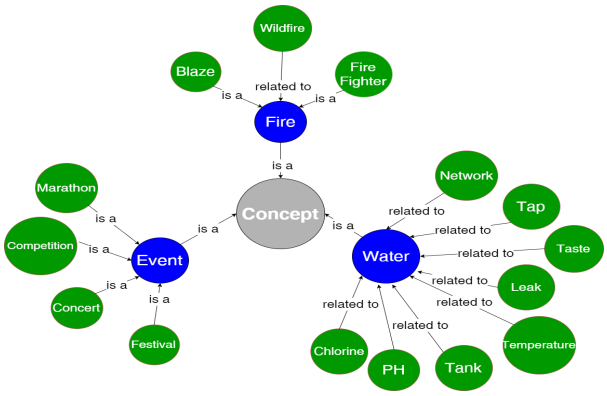

FIG. 2 – *Aperçu de l'ontologie d'extraction*

En combinant les concepts et les propriétés avec les prédicats, nous pouvons créer une ontologie expressive telle que celle de la figure 2 utilisée pour le cas d'utilisation des fuites d'eau. Ce type de structure est plus expressif qu'une liste classique de mots-clés de par sa modularité et son extensibilité.

3.2 Extraction de Résumés

Après avoir récupéré les événements pertinents des différentes sources de données en se basant sur l'ontologie de concepts et de propriétés, l'étape suivante consiste à extraire des résumés significatifs des événements en suivant le processus décrit dans la Figure 3.

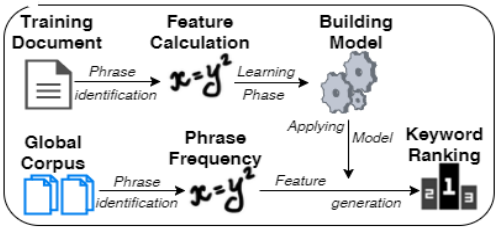

FIG. 3 – *Processus d'extraction de résumés*

Le prétraitement initial concerne le nettoyage du texte en entrée, l'identification des candidats potentiels et, enfin le tri ainsi que l'harmonisation des majuscules/minuscules. Les fichiers

en entrée sont filtrés pour régulariser le texte et déterminer les limites des phrases, puis interviennent le fractionnement en tokens ainsi que quelques opérations de nettoyage (*i.e.*, suppression des apostrophes ou séparation de certaines expression en plusieurs mots). Ensuite, nous considérons toutes les sous-séquences générées afin de pouvoir déterminer celles qui conviennent en tant que phrases complètes et compréhensibles. Pour augmenter la précision, nous utilisons une liste de mots français contenant plus de 500 entrées dans différentes classes syntaxiques (*e.g.*, conjonctions, articles, particules, etc.). Nous trions et harmonisons tous les mots via une méthode itérée très utilisée dans le domaine du TLN Lovins (1968), le processus est répété jusqu'à ce qu'il n'y ait plus d'amélioration possible.

Quant au traitement principal, il est relatif au calcul de deux valeurs distinctes pour chaque phrase candidate : la fréquence des phrases dans le texte d'entrée par rapport à sa rareté dans l'utilisation générale et la première occurrence, qui correspond à une distance dans le texte d'entrée indiquant la première apparition de la phrase. Ces deux valeurs sont converties en données nominales pour faciliter le processus de d'apprentissage automatique et une table de discrétisation pour chacune des valeurs est dérivée des données d'entraînement. Enfin, nous générons un modèle qui donne les scores pour chaque candidat et le classons en utilisant des techniques bayésiennes naïves Domingos et Pazzani (1997).

3.3 Pertinence des Résumés

Plusieurs travaux de recherche abordent la question du résumé automatique Ellouze et al. (2017). Dans notre cas, nous avons choisi une approche basée sur une similarité distribuée qui compare le contenu d'entrée et le résumé. Nous considérons qu'un bon résumé devrait être caractérisé par une faible divergence entre les distributions de probabilité des mots en entrée et le résumé généré, et par une forte similitude avec l'entrée. À cette fin, nous avons utilisé deux mesures complémentaires : la divergence de Kullback Leibler et la divergence de Jensen Shannon. Tout d'abord, les mots en entrée et dans le résumé sont triés et segmentés avant tout calcul. Ensuite, nous calculons les deux mesures :

Divergence de Kullback Leibler (KL) : Elle correspond au nombre moyen de bits utilisés pour le codage d'échantillons appartenant à P en utilisant une autre distribution Q, approximative de P. Elle est donnée par :

$$D_{\mathrm{KL}}(P\|Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)}.$$

Dans notre cas, les deux distributions de probabilités sont estimées à partir du texte en entrée et du résumé. Étant donné que la divergence de KL n'est pas symétrique, les divergences du texte en entrée et de son résumé sont introduites en tant que mesures. En outre, nous effectuons un lissage simple via une fonction d'approximation qui capture des comportements spécifiques tout en excluant le bruit et d'autres nuisances à faible échelle.

Divergence de Jensen Shannon (JS) : Elle s'appuie sur le fait que la distance entre deux distributions ne peut pas être trop éloignée de la moyenne des distances de leur distribution moyenne. Elle est donnée par la formule suivante :

$$\mathrm{JSD}(P \parallel Q) = \frac{1}{2} D(P \parallel M) + \frac{1}{2} D(Q \parallel M) \; avec \; M = \frac{1}{2}(P + Q) \tag{1}$$

Contrairement à la divergence de KL, la divergence de JS est symétrique et toujours définie. Nous calculons à la fois les versions lissées et non lissées de ces divergences en tant que résultats du résumé. La dernière étape est d'utiliser la sortie de ces deux fonctions pour classer les résumés extraits et ne conserver que ceux ayant le meilleur résultat de résumé (*i.e.*, les divergences les plus faibles).

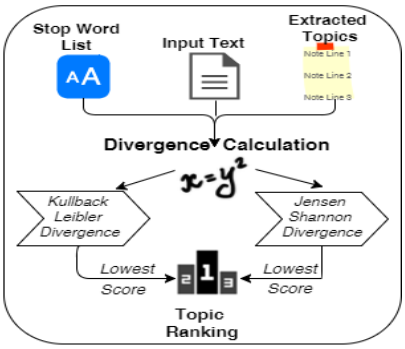

FIG. 4 – *Processus d'estimation de la pertinence du résumé*

3.4 Analyse Sentimentale

Au cours de la dernière décennie, l'analyse du sentiment a connu un développement exponentiel, de nombreuses solutions ont été proposées sur diverses technologies A. Collomb et Brunie (2014). Nous proposons dans cette section une approche simple et efficace mêlant des outils variés dont le toolkit fourni par Stanford CoreNLP Manning et al. (2014).

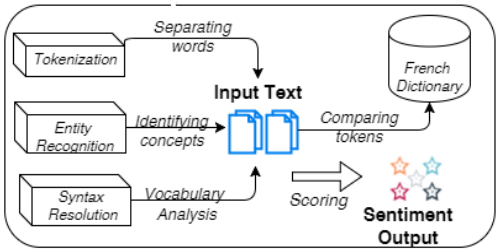

FIG. 5 – *Processus d'Analyse sentimentale*

Avant d'appliquer le modèle, nous devons effectuer plusieurs étapes de prétraitement qui améliorent la précision du score final en sortie. Les trois étapes réalisées ici sont **(a) la Tokenisation** (séparation du texte en une séquence de tokens et division chaque séquence en phrases significatives, **(b) la Reconnaissance d'entités** (inférence d'informations sur le genre puis annotation en tant que personnes, emplacements, organisations, nombres, etc) et **(c) la Résolution syntaxique** (recherche de dépendances grammaticales et utilisation d'un dictionnaire français).

Après la phase de prétraitement, nous appliquons le modèle de composition sur arbres basé sur une approche d'apprentissage profond. Il s'appuie sur les nœuds d'un arbre binarisé pour chaque phrase, y compris en particulier le nœud racine, chaque nœud étant annoté d'un score de sentiment. Afin de saisir le sentiment d'un texte en entrée, un modèle de réseau de neurones récursifs (*i.e.*, Recursive Neural Tensor Network ou RNTN) est construit en fonction des caractéristiques des phrases en entrée. Cette approche est inspirée des modèles récursifs et profonds développés par l'équipe Stanford Richard (2013).

3.5 Correspondance des Résumés

L'objectif des différentes étapes du module analytique est d'extraire les événements uniques les plus pertinents en les annotant avec un résumé expressif. Le système permet d'éviter de stocker des événements en doublons faisant référence à la même occurrence. Pour chaque événement extrait, le système proposera une liste de résumés potentiels basés sur une approche bayésienne. Ensuite, ces résumés seront classés en utilisant les divergences les plus faibles (*i.e.*, divergence KL et divergence JS) afin d'évaluer leur précision. Parmi les mieux classés, nous vérifierons s'ils disposent du même sentiment (*i.e.*, positif, neutre ou négatif). Si deux résumés sélectionnés au cours de ce processus ont le même score de pertinence et le même sentiment, nous supposons alors qu'ils se réfèrent au même événement. Par conséquent, nous concluons que ces événements sont des doublons et nous ne conserverons que le contenu d'un seul. Le processus global de notre module analytique est détaillé dans la figure6.

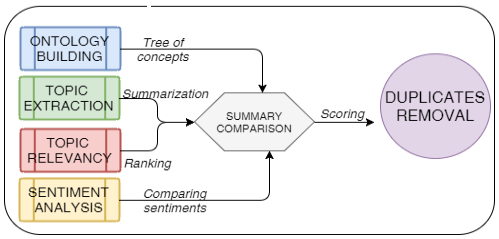

FIG. 6 – *Processus de correspondance des résumés*

Même si ce module fournit des fonctions puissantes pour filtrer les événements uniques et pertinents, une dimension spatiale est nécessaire afin de parfaire la contextualisation de l'anomalie détectée. Cette partie sera expliquée dans la section suivante.

4 Profilage Géographique

Afin de pouvoir établir la pertinence des événements détectés comme origine potentielle d'une anomalie, et pour pouvoir ajuster le score de probabilité qui leur est attribué, Scouter se base sur un système de profilage géographique Lhez et Curé (2016) ; L'objectif est de pouvoir déterminer la composition des secteurs de consommation étudiés en termes de terrain.

Le profilage est réalisé à partir de données cartographiques provenant d'OpenStreetMap Haklay et Weber (2008), un projet international sous licence libre. Le programme établit à partir des informations sur les secteurs de consommation les données à extraire, en construisant

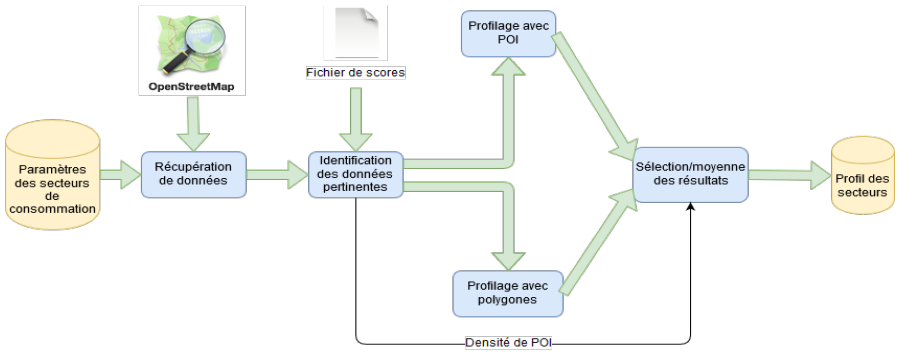

FIG. 7 – *Architecture du système de profilage*

une *bounding box* adaptée. Un fichier de configuration est également fourni au programme, afin de pouvoir établir quelles sont les grandes catégories de terrain à considérer, et quels tags définis par OSM appartiennent à ces catégories. A chaque tag est attribué une note, qui correspond à sa pertinence pour décrire la catégorie de terrain à laquelle il appartient. Le fichier, au format JSON, est donc organisé en une hiérarchie de catégories, incluant les tags en tant que feuille, mais aussi les différentes classes définies par OSM ; cela permet à l'utilisateur de créer ou modifier la configuration du profilage si nécessaire. Un exemple issu du fichier de scores pour le cas d'usage de Waves est fourni en figure 8.

```
"amenity": {
        "tourism": {
                "entertainment": {
                        "arts_centre": 0.8,
                        "casino": 1.0,
                        "community_centre": 0.8,
                        "fountain": 0.3,
                        "gambling": 1.0,
                        "planetarium": 0.8,
                        "theatre": 1.0}
        }
```

FIG. 8 – *Extrait de l'arborescence du fichier de configuration du profilage de Waves*

A partir de ces informations, deux profilages distincts sont réalisés : à partir des points d'intérêt (POI) et des polygones. La première méthode se base sur les *nodes* récupérés d'OSM, ainsi que sur les notes attribuées aux tags dans le fichier de configuration. Pour chaque mot clef identifié à la fois au sein des données cartographiques et au sein de la classification du fichier, on ajoute la note qui lui est attribuée au score total de la catégorie de terrain qu'il représente. Les catégories les plus représentées dans le secteur auront ainsi un score plus important, et il suffit ensuite d'un simple calcul de proportions finales pour obtenir la répartition finale. Cette méthode est très adaptée pour identifiées les zones denses en POI. La deuxième méthode se base sur l'utilisation des polygones pour établir la répartition. Le procédé est le même que précédemment pour l'identification des tags pertinents ; toutefois, à la place des notes attribuées arbitrairement dans le fichier de configuration, on calcule à la place la surface des polygones.

La répartition finale se calcule de la même manière, avec des scores pour chaque catégorie obtenus différemment, parfaits pour les secteurs riches en polygones.

Chaque méthode de profilage convient donc mieux à un cas précis, et il s'avère qu'elles sont également complémentaires. En effet, les zones géographiques comportant beaucoup de POI sont souvent dépourvues de polygones, car il s'agit de secteurs très divers, sans parcelles de terrain uniforme. A l'inverses, les secteurs riches en polygones seront également la plupart du temps très pauvres en POI, car il s'agit de terrains sans éléments remarquables. Ainsi, dans certains cas, les méthodes peuvent être sélectionnées, voire adaptées pour obtenir un résultat plus précis. Pour cela, il suffit de calculer la densité (proportion) de POI au kilomètre carré à partir des données d'OSM : de la sorte, l'utilisateur peut choisir quel profilage est le plus adapté à ses besoins. En cas de densité moyenne, il peut être pertinent de calculer la moyenne des résultats des profilages pour ajuster les proportions.

L'ensemble du système de profilage est donc configurable à partir du fichier de score, ce qui permet d'effectuer des ajustements. Par ailleurs, il est parfaitement possible de réaliser sa propre méthode de combinaison des profilage si l'approche générique ne convient pas.

5 Évaluation

Dans cette section, nous évaluons la performance du système sur plusieurs dimension tant quantitatives que qualitatives.

5.1 Media Analytics

Lors de cette expérimentation, nous avons collecté des flux durant 9 heures depuis des sources telles que Facebook, Twitter, des feeds RSS, Open Agenda, DBPedia et Open Weather Map. Notre cible géographique était l'agglomération de Versailles. Les paramètres utilisés pour chaque source sont présentés dans la Table 1. Par exemple, en utilisant l'API de streaming de Twitter, nous récupérons les flux de la zone géographique de Versailles mais aussi depuis les comptes *e.g.*, @Versailles et @monversailles. Les mots clés utilisés pour requêter ces tweets sont associés à 12 concepts de l'ontologie pour lesquels un score de pertinence est associé.

Source	Fréquence (heures)	Pages & Comptes	Concepts & Scores
Facebook	12	Mon Versailles, Versailles Officiel, Public Events	meter :1
Twitter	Streaming	@Versailles, @monversailles, @prefet78 #sdis78	damage :10 concert :10 spray :1, fire :10 water :10, blaze :1 wildfire :10, flow :5 tank :1, chlore :5 pressure :5
Open Agenda	24		
Open Weather Map	4		
DBpedia	24		
RSS News Papers	12	Le Parisien, 78 Actu, versailles.fr, Sdis78 yvelines.gouv.fr	

TAB. 1 – *Sources de Données & Scores des Concepts*

Performance du système : Deux métriques permettent de déterminer la performance de Scouter pour les fonctions les plus demandeuses en ressources. Le tableau 2 montre le temps moyen nécessaire pour annoter tous les événements collectés, il est calculé en divisant la somme des temps de scoring pour chacun des événements par le nombre d'événements collectés. Il montre également le temps d'entraînement pour l'algorithme d'extraction de résumés

visant à construire le modèle approprié. Nous pouvons voir que Scouter reste performant malgré un nombre relativement important d'événements traités par le système sans pour autant subir une panne ou un retard.

Mesure	Temps en Millisecondes
Temps Moyen de Traitement	7.43
Temps d'Entrainement Extraction de Résumés	474

TAB. 2 – *Temps de Traitement Scouter*

Qualité des événements collectés : Nous avons considéré notre cas d'utilisation sur les fuites d'eau durant l'année courante (2017). Notre système a tourné durant 9 heures pour collecter les événements de diverses sources du web, en utilisant l'ontologie explicitée dans la section et les scores assignés dans le tableau . L'exploitant du réseau d'eau potable a fourni l'horodatage et l'emplacement de toutes les anomalies rapportées en 2017 dénombrées à 15 au total. À partir de la base de données, nous avons récupéré tous les événements stockés correspondant à l'horodatage et l'emplacement de chaque anomalie et les avons présentés à cinq experts du domaine. Pour chaque événement, on leur a demandé d'estimer si cet événement pouvait fournir une explication pertinente à l'anomalie signalée. Une contrainte a été imposée, la réponse devait être se borner à "oui" ou "non," afin de simplifier l'interprétation des résultats.

Evaluateur	Evénements														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	×	✓	×	✓	✓	✓	×	✓	×	×	✓	×	×	×	×
2	×	✓	×	✓	✓	×	×	✓	×	✓	✓	×	×	×	×
3	×	✓	×	✓	✓	×	✓	×	×	✓	×	✓	✓	×	×
4	×	✓	×	✓	✓	×	×	✓	×	✓	×	×	✓	×	×
5	×	×	×	✓	×	×	×	✓	×	×	✓	×	×	×	×

TAB. 3 – *Évaluation des Experts de Domaine*

Pour évaluer la fiabilité de l'annotation, nous avons utilisé la mesure kappa de Fleiss Fleiss et al. (1971). Il s'agit d'une mesure statistique visant à évaluer la fiabilité de l'accord entre un certain nombre d'évaluateurs lors de l'attribution d'étiquettes à des sujets catégoriels. Cette mesure est exprimée par l'équation ci-dessous dont les résultats sont calculés pour un scénario avec 5 évaluateurs :

$$kappa = \frac{\bar{P} - \bar{P}_e}{1 - \bar{P}_e} = \frac{0.84 - 0.5256888889}{1 - 0.5256888889} = 0.6626686657$$

D'après le tableau d'interprétation des valeurs kappa Landis et Koch (1977), les évaluateurs ont un accord substantiel sur les événements annotés comme pouvant fournir une explication pertinente pour une anomalie de fuite d'eau. Par conséquent, Scouter a été assez efficace pour sélectionner les événements les plus pertinents.

5.2 Profilage Géographique

L'évaluation de la précision du profilage géographique a été réalisée à partir de divers échantillons de données fournis par notre client. Nous présentons ici un exemple basé sur la

zone de Versailles en France. Nous allons présenter les résultats des deux méthodes de profilage pour chaque secteur de consommation, et détailler les performances de chaque méthode du programme.

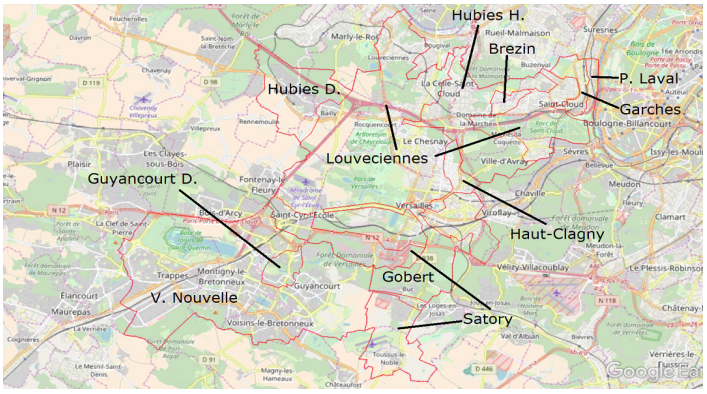

FIG. 9 – *Secteurs de consommations de Versailles, superposés aux données d'OpenStreetMap*

La figure 9 donne un aperçu de l'organisation du système de profilage. Le profilage utilise 5 types de terrains différents : résidentiel, agricole, naturel, industriel et touristique. Certaines zones sont plus facilement détectables en utilisant des POI, d'autres seront majoritairement représentées par des polygones. Les résultats finaux sont présentés dans le tableau 4. Nous avons utilisé la technique décrite en section 4, et nous réalisons la moyenne des deux méthodes si la densité est jugée moyenne. Une fois que nos ajustements finalisés, nous avons présenté nos résultats à un groupe d'experts du domaine pour recueillir leur avis (4ème colonne du tableau). Leurs évaluations sont majoritairement satisfaisantes, avec quelques remarques pour certains secteurs ne comportant pas de type de terrain majoritaire.

Zone	Catégorie	Densité de POI	Évaluation	Données OSM	Profilage	
					POI	Polygones
P. Laval	résidentielle	élevé	correct	5.4Mo	25ms	605ms
V. Nouvelle	rés. - nat.	moyen	correct	53.8Mo	282ms	630ms
Hubies D.	rés. - agr.	faible	nuancé	5.8Mo	13ms	30ms
Brezin	rés. - nat.	moyen	nuancé	3.1Mo	8ms	72ms
Guyancourt D.	rés. - nat.	moyen	correct	4.2Mo	13ms	50ms
Louveciennes	rés - tour.	élevé	correct	123.2Mo	1118ms	1290ms
Hubies H.	naturelle	faible	correct	37.15Mo	163ms	180ms
Haut-Clagny	résidentielle	élevé	correct	8.6Mo	21ms	32ms
Garches	touristique	élevé	correct	7.0Mo	205ms	46ms
Gobert	naturelle	faible	correct	15.4Mo	36ms	105ms
Satory	industrielle	faible	nuancé	32.5Mo	103ms	215ms

TAB. 4 – *Évaluation des résultats finaux*

Les dernières colonnes du tableau 4 détaillent les performances pour chaque méthode de profilage. La taille des données à télécharger dépend de la surface du secteur. Les données des polygones sont généralement plus volumineuses que celles des POI, même quand ces dernières sont plus nombreuses, leur représentation étant plus complexe. Ainsi, la méthode de profilage par polygone est généralement bien plus longue que celle des POI. Le temps d'exécution peut donc être potentiellement long pour des zones vastes, mais le profilage des secteurs ne dépend

d'aucune des données du flux reçu par Scouter, et peut donc être exécuté hors ligne pour ne pas impacter les performances.

Références

A. Collomb, C. Costea, D. J. O. H. et L. Brunie (2014). A study and comparison of sentiment analysis methods for reputation evaluation. Technical report.

Adam Berger, S. D. P. et V. D. Pietra (1996). A maximum entropy approach to natural language processing. *Computational Linguistics-MIT*, 22–1.

de S Sirisuriya, S. (2015). A comparative study on web scraping. *International Research Conference*.

Domingos, P. et M. Pazzani (1997). On the optimality of the simple bayesian classifier under zero-one loss. *Machine Learning*.

Ellouze, S., M. Jaoua, et H. Belguith (2017). Machine learning approach to evaluate multilingual summaries. In *Proceedings of the MultiLing 2017 Workshop*. Association for Computational Linguistics.

Fleiss, J. et al. (1971). Measuring nominal scale agreement among many raters. *Psychological Bulletin 76*.

Haklay, M. et P. Weber (2008). Openstreetmap : User-generated street maps. *IEEE Pervasive Computing*, 12–18.

Landis, J. R. et G. G. Koch (1977). The measurement of observer agreement for categorical data. *Biometrics*.

Lhez, J. et O. Curé (2016). Profilage sémantique et probabiliste de zones géographiques.

Lovins, J. (1968). Development of a stemming algorithm. *Mechanical Translation and Computational Linguistics*.

Manning, C. D., M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, et D. McClosky (2014). The stanford corenlp natural language processing toolkit. Association for Computer Linguistics.

Richard, S. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. Association for Computational Linguistics.

Summary

Anomaly detection is a key feature of applications processing singularities using IoT sensor measures. To guarantee high quality detections, meta-data providing spatio-temporal contexts on sensor measures are needed. In this paper, we introduce Scouter, a generic tool that helps in capturing, analyzing, scoring and storing the contextual information of a given application domain. The process depends on a semantic-based approach that exploits ontologies to score the relevancy of contextual information. The paper provides details on the system's architecture, describes its components and evaluates the performance based on real-world datasets.