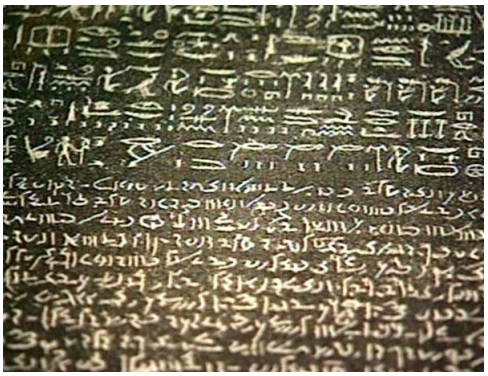
TextMine '21

Atelier sur la Fouille de Textes



Organisateurs :

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

Organisé conjointement à la conférence EGC
(Extraction et Gestion des Connaissances)
le 26 janvier 2021 à Montpellier

Editeurs :

Pascal Cuxac - INIST - CNRS
2 allée du Parc de Brabois, CS 10310, 54519 Vandoeuvre les Nancy Cedex
Email : pascal.cuxac@inist.fr

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

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

———

Publisher:

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

Lannion, France, 2021

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
V. Lemaire — Orange Labs
C. Lopez — Emvista

Membres du comité de lecture

Le Comité de Lecture est constitué de:

Guillaume Cabanac (IRIT, Toulouse)

Mariane Clausel (Université de Lorraine, Nancy)

Vincent Claveau (IRISA, Rennes)

Kevin Cousot (Emvista, Montpellier)

Natalia Grabar (STL - Lille3, Lille)

Sonia Le Meitour (Orange Labs, Lannion)

Denis Maurel (Université F. Rabelais, Tours)

David Reymond (Université de Toulon, Toulon)

TABLE DES MATIÈRES

FlauBERT : des modèles de langue contextualisés pré-entraînés pour le français

Didier Schwab*

*Univ. Grenoble Alpes, CNRS, Laboratoire d'Informatique de Grenoble
didier.schwab@univ-grenoble-alpes.fr

En 2018, l'introduction de représentations linguistiques profondes contextuelles, obtenues à partir de textes bruts, a conduit à un changement de paradigme pour plusieurs tâches du TALN. Alors que les approches fondées sur des représentations continues telles que Word2vec (Mikolov et al., 2013) ou GloVe (Pennington et al., 2014) apprennent un vecteur unique pour chaque mot, les modèles introduits alors produisent des *représentations contextuelles* qui dépendent de la séquence de mots d'entrée complète. Initialement fondées sur des réseaux neuronaux récurrents (Dai et Le, 2015; Ramachandran et al., 2017; Howard et Ruder, 2018; Peters et al., 2018), ces approches ont peu à peu intégré des modèles *Transformer* (Vaswani et al., 2017) comme c'est le cas pour GPT (Radford et al., 2018), BERT (Devlin et al., 2019), XLNet (Yang et al., 2019), RoBERTa (Liu et al., 2019), ALBERT (Lan et al., 2019) et T5 (Raffel et al., 2019). L'utilisation de ces modèles pre-entraînés a permis des avancées de l'état-de-l'art pour de nombreuses tâches du TALN. Cependant, ceci a surtout été montré pour l'anglais, même si des variantes multilingues sont également disponibles, prenant en compte plus d'une centaine de langues dans un seul modèle : mBERT (Devlin et al., 2019), XLM (Lample et , 2019), XLM-R (Conneau et al., 2019). Nous décrivons ici notre méthodologie pour construire et partager FlauBERT (**F**rench **La**nguage **U**nderstanding via **B**idirectional **E**ncoder **R**epresentations from **T**ransformers), un modèle BERT pour la français.

Nous proposons aussi un référentiel d'évaluation nommé FLUE (**F**rench **L**anguage **U**nderstanding **E**valuation) similaire au benchmark GLUE (Wang et al., 2018) pour l'anglais et montrons que les modèles FlauBERT surpassent ou obtiennent des résultats similaires aux autres modèles de langue sur le français.

FlauBERT et FLUE sont disponibles librement en ligne pour la communauté scientifique, l'industrie et plus généralement pour tous les curieux cherchant à explorer les possibilités aujourd'hui offertes par l'apprentissage profond sur le français et leurs secrets.

Références

Conneau, A., K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, et V. Stoyanov (2019). Unsupervised cross-lingual representation learning at scale. *arXiv preprint https ://arxiv.org/abs/1911.02116arXiv :1911.02116*.

Dai, A. M. et Q. V. Le (2015). Semi-supervised sequence learning. In *Advances in neural information processing systems*, pp. 3079–3087.

Devlin, J., M.-W. Chang, K. Lee, et K. Toutanova (2019). Bert : Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186.

Howard, J. et S. Ruder (2018). Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, pp. 328–339.

Lample, G. et A. (2019). Cross-lingual language model pretraining. In *Advances in neural information processing systems*.

Lan, Z., M. Chen, S. Goodman, K. Gimpel, P. Sharma, et R. Soricut (2019). Albert : A lite bert for self-supervised learning of language representations. *arXiv preprint https ://arxiv.org/abs/1909.11942arXiv :1909.11942*.

Liu, Y., M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, et V. Stoyanov (2019). Roberta : A robustly optimized bert pretraining approach. *arXiv preprint https ://arxiv.org/abs/1907.11692arXiv :1907.11692*.

Mikolov, T., I. Sutskever, K. Chen, G. Corrado, et J. Dean (2013). Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, USA, pp. 3111–3119. Curran Associates Inc.

Pennington, J., R. Socher, et C. D. Manning (2014). Glove : Global vectors for word representation. In *In EMNLP*.

Peters, M. E., M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, et L. Zettlemoyer (2018). Deep contextualized word representations. In *Proceedings of NAACL-HLT*, pp. 2227–2237.

Radford, A., K. Narasimhan, T. Salimans, et I. Sutskever (2018). Improving language understanding by generative pre-training. *Technical report, OpenAI*.

Raffel, C., N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, et P. J. Liu (2019). Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint https ://arxiv.org/abs/1910.10683arXiv :1910.10683*.

Ramachandran, P., P. Liu, et Q. Le (2017). Unsupervised pretraining for sequence to sequence learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 383–391.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, et I. Polosukhin (2017). Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008.

Wang, A., A. Singh, J. Michael, F. Hill, O. Levy, et S. Bowman (2018). GLUE : A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP : Analyzing and Interpreting Neural Networks for NLP*, Brussels, Belgium, pp. 353–355. Association for Computational Linguistics.

Yang, Z., Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, et Q. V. Le (2019). Xlnet : Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*.

Extraction d'informations spécifiques à partir de textes avec peu de textes d'apprentissage

Bénédicte Goujon

THALES Research & Technology France, 1 avenue Augustin Fresnel,
91767 Palaiseau cedex
benedicte.goujon@thalesgroup.com

Résumé. De nombreux outils statistiques sont aujourd'hui disponibles pour traiter certaines tâches liées à l'extraction d'informations à partir de textes. Cependant, il n'existe pas à notre connaissance d'outil permettant d'extraire des informations dans des contextes pauvres en corpus pré-annotés. Cet article présente la plateforme STRASS qui vise à supporter l'annotation automatique et l'extraction d'informations à partir de textes en combinant des approches statistiques et des approches symboliques via l'apprentissage automatique de patrons linguistiques.

1 Introduction

Alors que l'extraction d'informations semble être devenue une tâche facile, grâce au développement et au succès des méthodes d'apprentissage à base de statistiques et de réseaux de neurones, il n'existe toujours pas sur le marché d'outil visant l'extraction d'informations dans des contextes métiers spécifiques où peu de ressources textuelles annotées peuvent être disponibles. La plateforme STRASS vise à palier ce manque en combinant des outils statistiques existants et un apprentissage de patrons linguistiques pour les informations complexes comme les relations entre entités et les argumente d'événements. Après avoir décrit des besoins liés à l'extraction d'informations et présenté un état de l'art sur les outils disponibles et les recherches en cours en lien avec des approches statistiques et des approches symboliques, nous présentons en détail les caractéristiques de la plateforme STRASS ainsi qu'un exemple d'utilisation.

2 L'extraction d'informations

2.1 Présentation

Nous distinguons quatre besoins liés à l'extraction d'informations : l'extraction d'entités nommées génériques, l'extraction d'entités et événements spécifiques, l'extraction de relations entre entités et enfin l'extraction d'arguments reliant des événements et des entités. Les approches statistiques sont actuellement très efficaces pour l'extraction d'entités nommées génériques, avec des outils proposant des modèles d'extraction prédéfinis, appris sur des corpus

pré-annotés existants. Les approches statistiques peuvent aussi être utilisées pour l'extraction d'entités plus spécifiques sous réserve de disposer de grands corpus pré-annotés pour la phase d'apprentissage d'un modèle spécifique. Dans des contextes où ces corpus ne sont pas disponibles, de même que dans les cas où des relations entre entités et des arguments d'événements sont recherchés, aucun outil statistique disponible actuellement ne semble fournir de solutions. Les approches symboliques, basées sur la construction de patrons par des linguistes pour l'annotation de relations et d'arguments, se heurtent au besoin de compétences linguistiques pour l'écriture des patrons.

2.2 État de l'art

Pour l'extraction automatique d'entités nommées, il existe plusieurs outils basés sur des approches statistiques, offrant des modèles sur étagère pour l'annotation d'entités génériques et permettant d'effectuer un apprentissage sur des corpus volumineux annotés pour la gestion d'entités spécifiques. Les principaux outils sont spaCy [1] (qui propose 3 modèles pour l'anglais et autant pour le français), NLTK [2], OpenNLP [3] ou coreNLP (Manning et al. (2014)). Pour l'extraction d'informations plus complexes comme les relations et les arguments, il est possible d'utiliser des approches symboliques à base de règles, où des patrons sont construits manuellement par des linguistes, comme proposé par les outils Unitex [4] ou Gate [5]. Les patrons linguistiques définissent des contraintes linguistiques pour, par exemple, identifier des arguments d'événements. Ainsi, pour un événement de type Attaque, un patron linguistique permettra de repérer la description de la victime en s'appuyant sur l'identification d'une expression verbale (”être victime” ou ”attaquer”), d'un type d'entité (Personne) et de la position ou catégorie grammaticale de cette description (sujet ou objet direct). Par ailleurs, il existe des outils d'annotation manuelle, comme prodigy [6], BRAT [7] ou WebAnno [8], qui permettent l'annotation manuelle de textes par des experts métier pour la construction des corpus d'apprentissage. Ces trois outils permettent l'annotation d'entités, d'événements, de relations et d'arguments. L'annotation de relations et dépendances est une nouvelle capacité intégrée à la version 1.10 de Prodigy en juin 2020.

L'acquisition manuelle de patrons linguistiques étant une tâche fastidieuse, dès 1993 des travaux de recherche ont porté sur l'apprentissage automatique de patrons linguistiques, avec l'outil AutoSlog d'Ellen Riloff (Riloff (1993)). AutoSlog permettait l'acquisition de patrons d'événements à partir d'un ensemble de patrons linguistiques génériques prédéfinis et d'un corpus annoté manuellement, pour l'extraction d'événements terroristes. L'outil ExDisco, présenté en 2000 (Yangarber et al. (2000)), permettait d'apprendre des patrons linguistiques à partir d'un petit ensemble de patrons initial. Dans le même temps, des campagnes d'évaluation des tâches d'extraction d'informations comme MUC (Grishman et Sundheim (1996), Hirschman (1998)) puis ACE de 1999 à 2008 (Doddington et al. (2004)) ont débouché sur la création de corpus annotés volumineux, contribuant à l'essor de nouvelles méthodes d'apprentissage, à

1. https ://spacy.io
2. http ://www.nltk.org/
3. https ://opennlp.apache.org/
4. https ://unitexgramlab.org/fr
5. https ://gate.ac.uk/
6. https ://prodi.gy
7. http ://brat.nlplab.org
8. https ://webanno.github.io/webanno/

base de statistiques ou de réseaux de neurones, pour l'extraction d'événements et leurs arguments à partir de textes.

De notre côté, de 2004 à 2010 nous avons développé le prototype SemPlus (Goujon (2008)) d'acquisition de patrons linguistiques pour l'extraction d'informations à partir de textes. Cet outil permettait de construire des patrons d'événements reliant deux entités, pour par exemple repérer des déplacements reliant des personnes et des lieux ou des rencontres entre deux personnes. SemPlus, développé en Java, s'appuyait sur des dictionnaires prédéfinis contenant des noms d'entités et sur l'outil Intex (Silberztein (2000)) pour l'application des dictionnaires et des patrons sur des textes.

Aujourd'hui de nombreuses méthodes statistiques permettent de résoudre un grand nombre de problèmes d'extraction d'informations, en particulier des méthodes s'appuyant sur les réseaux de neurones, par exemple celle présentée par Kodelja et al. (Kodelja et al. (2019)) à base de CNN (*convolutional neural networks*) qui vise la détection d'événements en tenant compte d'un contexte plus étendu que la phrase, mais celles-ci nécessitent l'existence de vastes corpus d'apprentissage pré-annotés. Afin de faciliter la construction de ces corpus d'apprentissage, certains travaux de recherche s'intéressent à l'expansion des données d'apprentissage à partir de textes non annotés manuellement, comme ceux de Liao et Grishman (Liao et Grishman (2011)) qui s'appuient sur une stratégie d'apprentissage actif afin d'alléger la tâche d'annotation manuelle, ou ceux plus récents de Wang et al. (Wang et al. (2019)) qui visent l'annotation automatique en recherchant les occurrences des termes principaux d'événements (*triggers*) dans des corpus non annotés. Pour toutes les approches, la difficulté consiste à trouver le meilleur compromis pour un besoin visé entre d'une part une bonne qualité des résultats avec un coût élevé de main d'oeuvre et d'autre part une automatisation plus importante associée à une moins bonne qualité.

3 La plateforme STRASS

3.1 Présentation

Aujourd'hui il n'existe pas à notre connaissance d'outil disponible pour l'apprentissage de modèles statistiques spécifiques à partir de faibles corpus annotés pour les entités, les événements et les relations. C'est pourquoi nous avons choisi de développer notre propre plateforme, nommée STRASS, qui s'appuie sur la combinaison d'approches statistiques et d'approches symboliques. La plateforme STRASS est centrée sur l'apprentissage de patrons linguistiques à partir de textes annotés manuellement par un expert métier, et a pour objectif de permettre la gestion de l'ensemble des tâches, de la construction du modèle ontologique jusqu'à l'affichage des textes annotés produits automatiquement ou l'export des informations extraites. Un atout de la plateforme STRASS est de viser un coût de main d'œuvre réduit, inférieur à celui des méthodes s'appuyant sur des patrons construits par des linguistes, et inférieur au coût des annotations manuelles nécessaires pour envisager l'utilisation d'approches statistiques. La qualité des résultats obtenus pourra être inférieure à celle d'approches concurrentes, comme celles à base de patrons construits par des linguistes, mais les résultats devront à terme être accompagnés d'explications, par exemple sous la forme de l'extrait de texte annoté manuellement ayant généré le patron à l'origine de chaque annotation, afin que l'outil soit accepté par les experts métier.

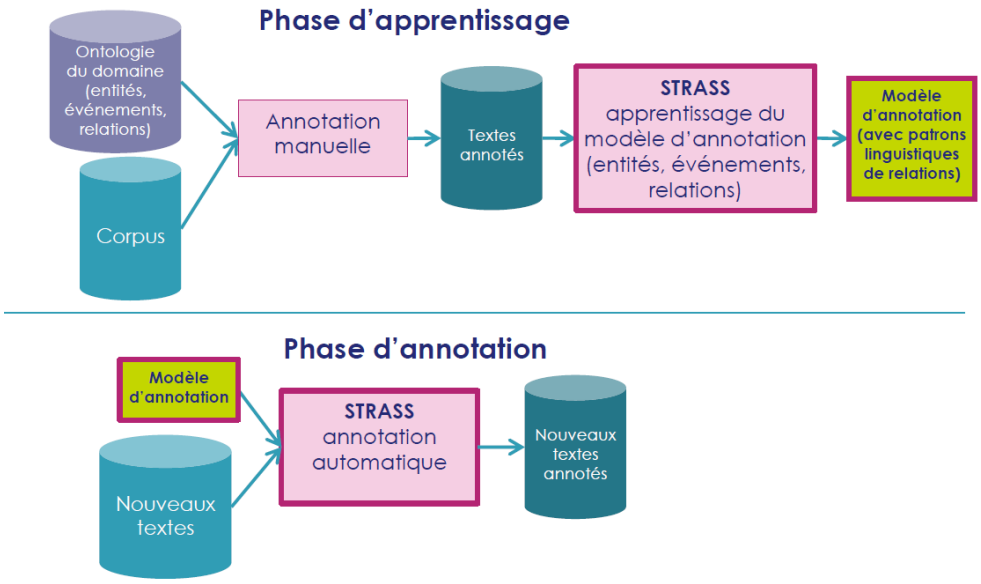


FIG. 1 – *Phases de la plateforme STRASS.*

Les spécificités de cette plateforme sont les suivantes :
— L'annotation des entités génériques est gérée par des modèles statistiques existants ;
— L'annotation des entités spécifiques s'appuie sur l'acquisition de dictionnaires fournis par l'expert métier ;
— L'annotation automatique de relations entre entités et d'arguments d'événements se fait à l'aide des patrons et dictionnaires acquis en phase d'apprentissage. Quelques textes annotés manuellement suffisent à obtenir des annotations automatiques, puisqu'un patron précis est généré pour chaque occurrence de relation dans le corpus d'apprentissage ;
— La plateforme STRASS peut être utilisée directement par l'expert métier, puisqu'aucune compétence linguistique n'est requise lors de l'annotation manuelle des textes.

Dans un premier temps, un modèle des concepts du domaine (entités, événements, relations), qui peut être une ontologie ou une simple liste de noms de concepts, est défini. Les relations sont de deux types : les relations entre entités et les relations de type argument qui relient des événements et des entités. Une fois les connaissances du domaine construites, l'utilisation de la plateforme STRASS se fait en deux phases principales, comme illustré figure 1.

La phase d'apprentissage consiste en l'annotation manuelle par un expert métier d'un ensemble réduit de textes représentatifs des concepts et relations visées. L'annotation se fait selon deux dimensions : l'annotation des entités et des événements d'une part, l'annotation des relations entre ces entités et événements d'autre part. Le modèle d'annotation obtenu à partir de cet ensemble de textes annotés contient des patrons linguistiques et des dictionnaires. Ce modèle peut être complété par des dictionnaires fournis par l'expert métier et par des patrons

linguistiques génériques prédéfinis.

Lors de la deuxième phase, le modèle d'annotation est appliqué sur de nouveaux textes du domaine pour leur annotation automatique. Tout d'abord, les analyses linguistiques génériques sont effectuées : découpage en mots, analyse grammaticale, annotation d'entités génériques. Puis, les dictionnaires et les patrons linguistiques spécifiques au domaine sont appliqués, selon les différentes contraintes qui les caractérisent. Deux résultats sont obtenus : les textes annotés d'une part, et les informations extraites des textes, correspondant aux annotations, d'autre part.

Lors de la première phase, l'annotation manuelle se fait à l'aide de l'outil WebAnno. Cet outil gère les corpus de textes à annoter, les noms des types d'entités et événements et les noms des types de relations selon deux couches d'annotation distinctes, et permet de saisir des annotations sur une IHM assez intuitive. Les textes annotés sont ensuite analysés dans la plateforme STRASS à l'aide d'un code spécifique qui s'appuie sur l'outil spaCy et qui génère des patrons linguistiques. Le format des patrons s'inspire de celui que spaCy utilise dans ses composants à base de règles, mais contrairement à ceux-ci, il permet de ne pas d'appuyer uniquement sur des séquences de mots consécutifs du texte et permet d'utiliser d'autres attributs que les neuf définis par défaut (POS, LEMMA, ORTH, SHAPE...).

L'annotation de nouveaux textes est alors réalisable à l'aide du modèle d'annotation acquis en phase d'apprentissage, avec à nouveau le support de spaCy pour les analyses génériques. Les textes annotés obtenus sont générés dans un format lisible pour l'expert métier grâce à l'outil WebAnno (voir figure 2) ainsi que dans un format JSON permettant la connexion avec l'outil de fusion d'informations InSyTo (Laudy (2017)), pour des traitements ultérieurs des informations extraites des textes.

3.2 Exemple

La plateforme a été testée sur un ensemble de 300 textes très courts en anglais tirés des descriptions du corpus Predicting Terrorism [9], portant sur le bilan d'attaques terroristes. Voici quelques données quantitatives de ce test :

- Modèle du domaine :
  - Entités : 18 types d'entité générique (gérés par spaCy), 1 type d'entité spécifique : WEA (*weapon*).
  - Evénements : 4 types d'événement spécifique : Attack, Injure, Murder, Die.
  - Relations : 2 types de relations entre entités (How-Many, Affiliation), 2 types d'arguments génériques (where, when), 4 types d'arguments spécifiques : attacker, deadVictim, injuredVictim, mean.
- Apprentissage sur 11 textes (13 phrases) :
  - Dictionnaires : 8 ajouts manuels de synonymes et 32 entrées acquises lors de l'apprentissage.
  - Patrons : 5 patrons génériques prédéfinis (where, when, How-Many), 3 patrons de relations entre entités acquis lors de l'apprentissage, 21 patrons d'arguments acquis lors de l'apprentissage.
- Annotation de 300 textes.
  - Nombre d'annotations produites : 2 871.
  - Entités annotées : 622 PERSON, 461 CARDINAL, 273 NORP, 74 WEA...

9. https ://www.kaggle.com/argolof/predicting-terrorism

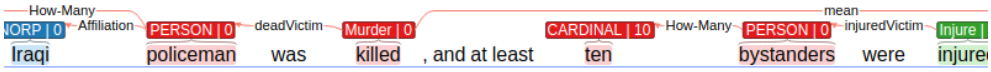


FIG. 2 – *Exemple de texte annoté automatiquement.*

— Evénements annotés : 214 Murder, 146 Injure, 19 Die, 18 Attack.
— Relations entre entités : 383 How-Many, 147 Affiliation. Arguments : 208 deadVictim, 84 InjuredVictim, 78 attacker, 41 mean. . .

La figure 2 montre un exemple de texte annoté automatiquement. La phrase initiale est la suivante : "*One Iraqi policeman was killed ,and at least ten bystanders were injured when a car bomb exploded outside a police station*". Les annotations produites contiennent des entités génériques de type NORP, PERSON et CARDINAL, une entité spécifique de type WEA ("*bomb*"), des événements de type Murder et Injure, une relation d'Affiliation, une relation How-Many et la détection de trois arguments : deadVictim, injuredVictim et mean.

4 Conclusion et perspectives

La plateforme STRASS, qui vise l'annotation et l'extraction d'informations spécifiques à un domaine sans nécessiter de corpus volumineux pré-annotés, est en cours de développement. De nombreux réglages restent à intégrer afin d'améliorer les résultats obtenus en tenant compte des nombreux contextes possibles pouvant exprimer des relations. Nous avons présenté un exemple de son utilisation et quantifié ses résultats. Nous envisageons de réaliser une évaluation plus qualitative des annotations produites sur notre corpus de test, ainsi que de tester notre approche sur des textes en lien avec des besoins réels, à l'aide d'experts métiers. L'évaluation qualitative sera complexe à mettre en oeuvre, car il sera utile d'évaluer non seulement le rappel et la précision, mais aussi la spécificité des entités, relations et événements visés. Par exemple, il sera intéressant de tenir compte de la variabilité des expressions associées à chaque concept, de la représentativité de cette variabilité dans le corpus d'apprentissage (complété par quelques synonymes saisis manuellement) et de la qualité de l'annotation statistique des éléments génériques produite par spaCy sur laquelle s'appuie l'annotation des relations et arguments. Enfin, il sera important de caractériser les cas où les annotations automatiques sont de mauvaises qualité afin d'accompagner l'expert métier dans sa compréhension des résultats obtenus automatiquement.

Une prochaine étape vise l'intégration des modules STRASS à la plateforme INCEPTION [10], qui contient l'outil d'annotation manuelle WebAnno ainsi qu'un onglet pour la gestion d'ontologie et un gestionnaire d'outils de recommandation (*recommenders*) pour la suggestion automatique d'annotations pour supporter la phase d'annotation manuelle. Enfin, certaines approches d'expansion des données d'apprentissage vont être étudiées pour une éventuelle intégration à notre plateforme afin d'améliorer l'annotation et l'extraction d'informations.

10. https ://inception-project.github.io/

Références

Doddington, G., A. Mitchell, M. Przybocki, L. Ramshaw, S. Strassel, et R. Weischedel (2004). The automatic content extraction (ace) program-tasks, data, and evaluation. *Proceedings of LREC 2*.

Goujon, B. (2008). Relation extraction in an intelligence context.

Grishman, R. et B. Sundheim (1996). Message understanding conference 6 : A brief history. Volume 96, pp. 466–471.

Hirschman, L. (1998). The evolution of evaluation : Lessons from the message understanding conferences. *Computer Speech & Language 12*, 281–305.

Kodelja, D., R. Besançon, et O. Ferret (2019). *Exploiting a More Global Context for Event Detection Through Bootstrapping*, pp. 763–770.

Laudy, C. (2017). Rumors detection on social media during crisis management.

Liao, S. et R. Grishman (2011). Using prediction from sentential scope to build a pseudo co-testing learner for event extraction. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, Chiang Mai, Thailand, pp. 714–722. Asian Federation of Natural Language Processing.

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

Riloff, E. (1993). Automatically constructing a dictionary for information extraction tasks. pp. 811–816.

Silberztein, M. (2000). Intex : an fst toolbox. *Theor. Comput. Sci. 231*, 33–46.

Wang, X., X. Han, Z. Liu, M. Sun, et P. Li (2019). Adversarial training for weakly supervised event detection. pp. 998–1008.

Yangarber, R., R. Grishman, P. Tapanainen, et S. Huttunen (2000). Automatic acquisition of domain knowledge for information extraction. *Proceedings of the 18th international conference on Computational linguistics*.

Summary

Many statistical tools are nowadays available to treat some tasks related to information extraction from texts. But no existing tool seems available to extract information in context with few pre-annotated corpora. This paper presents the STRASS plateform that aims to support the automatic annotation and information extraction from texts, combining statistical and symbolic approaches through the automatic learning of linguistic patterns.

Concevoir un assistant conversationnel de manière itérative et semi-supervisée avec le clustering interactif

Erwan Schild[*,**], Gautier Durantin[*], Jean-Charles Lamirel[**]

[*] Euro-Information Développements, Groupe Crédit-Mutuel
4 Rue Frédéric-Guillaume Raiffeisen 67000 Strasbourg,
prenoms.nom@e-i.com, https://www.e-i.com/
[**] LORIA, Université de Lorraine
615 Rue du Jardin-Botanique, 54506 Vandoeuvre-lès-Nancy,
prenoms.nom@loria.fr, https://www.loria.fr/

Résumé. La création d'un jeu de données nécessaire à la conception d'un assistant conversationnel résulte le plus souvent d'une étape manuelle et fastidieuse qui manque de techniques destinées à l'assister. Pour accélérer cette étape d'annotation, nous proposons une méthode de *clustering* interactif : il s'agit d'une approche itérative inspirée de l'apprentissage actif, reposant sur un algorithme de *clustering* et tirant parti d'une annotation de contraintes pour guider le regroupement des questions en une structure d'intentions. Dans cet article, nous exposons la méthodologie à mettre en oeuvre pour concevoir un assistant conversationnel opérationnel à l'aide du *clustering* interactif.

1 Introduction et enjeux

L'utilisation des assistants conversationnels (*chatbot*) explose car ces derniers permettent efficacement d'accéder à l'information avec des requêtes en langage naturel. Toutefois, leur création est encore fastidieuse en raison du manque de techniques permettant d'assister l'annotation du jeu de données nécessaire à leur entrainement. En effet, cette étape résulte le plus souvent d'un travail manuel et empirique possédant plusieurs défauts. On peut notamment citer le besoin de définir un modèle de catégorisation des données en intentions [1] ou encore la difficulté de maintenir cette modélisation abstraite sans introduire de biais ou d'ambiguïtés.

Pour assister l'humain dans cette tâche d'annotation, nous cherchons une alternative à l'approche manuelle en introduisant des initiatives de la machine. Nous nous intéressons plus particulièrement à la méthode du *clustering* interactif définie initialement par Gançarski et Wemmert (2007) et Lampert et al. (2019) pour la détection d'objets dans une image, une approche reposant sur la combinaison entre l'apprentissage actif et le *clustering* sous contraintes.

Néanmoins, le *clustering* interactif reste une méthode peu explorée dans le cadre du traitement du langage naturel. Dans Schild et al. (2021), nous avons proposé une première implémentation fonctionnelle cette technique, et nous nous intéressons désormais à l'intégration du *clustering* interactif dans le processus de conception d'un assistant conversationnel. Pour cet

1. Par exemple, «*Joue-moi du jazz !*» peut être modélisé par l'intention "*jouer de la musique*" et l'entité "*jazz*".

article, nous ferons abstraction des estimations de charge de travail pour discuter des impacts méthodologiques de cette nouvelle approche et définir les flux de validation associés. Ces estimations de charge seront réalisées dans une étude ultérieure mettant en oeuvre le protocole défini.

2 Clustering interactif

Principe général Le *clustering* interactif est une méthode semi-supervisée qui repose sur l'alternance successive entre une annotation de contraintes (*MUST-LINK*, *CANNOT-LINK*) et un *clustering* sous contraintes. L'objectif ainsi recherché est la création d'un cercle vertueux (cf. figure 1) pour améliorer itérativement la pertinence du *clustering* obtenu : un oracle suggère un ensemble de contraintes à annoter pour corriger efficacement le *clustering* issu de l'itération précédente, et le *clustering* exploite les contraintes annotées jusqu'à présent pour suggérer un nouveau partitionnement plus pertinent pour l'itération suivante.

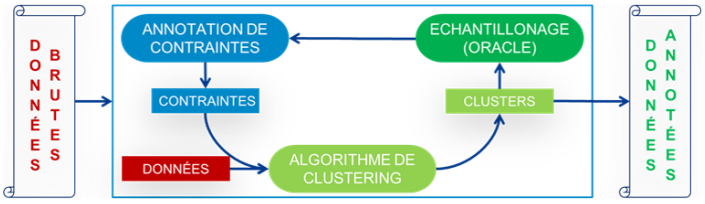


FIG. 1 – *Schéma représentant la boucle d'itérations du clustering interactif. Les étapes en bleu représentent l'initiative humaine, et les étapes en vert représentent l'initiative machine.*

Résultats obtenus Dans Schild et al. (2021), nous avons, à l'aide d'un corpus de données textuelles issues du domaine bancaire, confirmé la faisabilité technique d'une annotation avec cette méthode, nous permettant ainsi de déduire les propriétés suivantes :

- la définition d'un modèle d'organisation des données en intentions n'est plus un prérequis pour réaliser la phase d'annotation : en effet, cette structure d'intention émerge naturellement des contraintes annotées au cours des itérations, offrant ainsi un gain de temps majeur pour la conception d'un assistant ;
- avec l'utilisation de contraintes, l'annotation devient un mécanisme binaire centré sur la similarité des réponses à donner aux questions : l'annotateur n'a donc plus besoin d'avoir une connaissance globale de la structure d'intentions définie, ce qui réduit la complexité de sa tâche et minimise la possibilité d'introduire des contradictions ;
- la tâche d'annotation est désormais partagée entre l'homme et la machine : la charge de travail de l'annotateur est donc réduite car il n'intervient que pour fournir la dose d'information nécessaire pour améliorer la pertinence du *clustering* obtenu ;
- au cours des itérations, le partitionnement des données peut être efficacement corrigé grâce à l'annotation des contraintes adéquates : le choix de la méthode de sélection des données à annoter est donc très important ;
- comme les interactions successives permettent d'améliorer le partitionnement des données, il est possible d'obtenir un résultat pertinent malgré l'emploi d'algorithmes de

clustering simples : on peut donc privilégier la rapidité à la performance pour choisir la méthode de *clustering* car le mécanisme d'itérations comblera en partie cette lacune.

3 Intégration pour la conception d'un chatbot complet

Pour utiliser le *clustering* interactif dans le cycle de conception d'un assistant conversationnel, nous devons définir un mode d'emploi pour s'assurer de la pertinence du partitionnement des données que l'on obtient. Le protocole d'utilisation que nous proposons est schématisé dans la figure 2 et est détaillé ci-après. Ce dernier met en avant plusieurs pistes d'étude que nous devrons traiter ultérieurement afin de confirmer la viabilité de ce protocole.

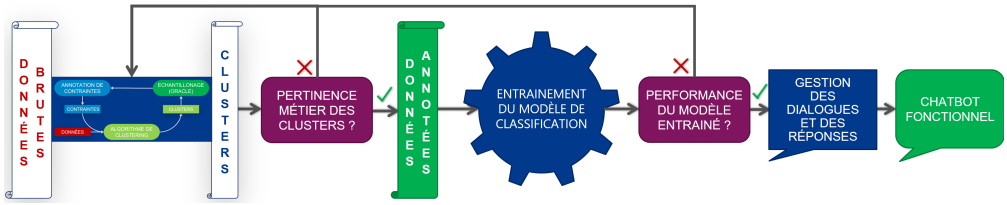


FIG. 2 – *Schéma représentant le protocole de mise en oeuvre d'un assistant conversationnel avec le clustering interactif. Les étapes en bleu foncé sont les actions à réaliser, celles en violet sont les évaluations intermédiaires et les objets en vert sont les livrables de la méthode.*

Annotation avec le *clustering* interactif. Bien que l'implémentation de notre méthode soit fonctionnelle pour le traitement de données textuelles, il reste quelques inconnues à résoudre pour la rendre pleinement opérationnelle.

Tout d'abord, on peut se demander comment estimer le nombre de *clusters* optimal. En effet, cette information est initialement inconnue et difficilement identifiable. Un *clustering* collaboratif pourrait représenter une piste pour se rapprocher de ce nombre optimal au cours des itérations, voire pour l'obtenir.

Ensuite, il faut être capable de prévenir ou résoudre un conflit d'annotation de contraintes entre les données. Ce problème est bien connu de l'apprentissage incrémental, et une étude approfondie de ce type d'apprentissage devrait permettre de le contourner.

Enfin, il demeure la question de l'exhaustivité de l'annotation. En effet, annoter toutes les contraintes entre les données est impensable en pratique compte tenu de la charge de travail nécessaire. Il peut être néanmoins pertinent de se contenter d'une annotation partielle et d'estimer si le jeu de données résultant est suffisamment fiable pour concevoir un assistant. Les paragraphes suivants détaillent les analyses qu'il est possible de mener en ce sens.

Analyse de la pertinence sémantique. Après plusieurs itérations du *clustering* interactif, nous pouvons nous interroger sur la pertinence sémantique des *clusters* obtenus : pouvons-nous associer une intention à ces *clusters* ? Comme l'annotation de contraintes se fait sur la base de similarité de la réponse (Schild et al., 2021), il est légitime de supposer que chaque

cluster doit en effet pouvoir être identifié par une réponse générale. Ce critère est fortement discriminant car il conditionne la connaissance dont l'assistant dispose. En conséquence, un *clustering* jugé trop peu pertinent demandera l'annotation de contraintes supplémentaires.

Analyse de la performance statistique. Si l'analyse de la pertinence des *clusters* est satisfaisante, nous pouvons alors nous intéresser aux contraintes techniques liées à la classification des intentions. En effet, la qualité de l'assistant dépend en grande partie de l'efficacité de cette détection, et il faut donc s'assurer de la performance du modèle entraîné, par exemple avec l'emploi d'un test K-folds et l'analyse d'une matrice de confusion. Si les performances ne sont pas satisfaisantes, l'analyse des dépendances entre *clusters* permettrait de définir si un remaniement manuel de la structure d'intention suffirait à corriger cette situation ou s'il faudrait ré-annoter d'autres contraintes pour identifier des intentions non reconnues jusque-là.

4 Conclusion

En suivant les perspectives que nous avons décrites, et après validation des critères de pertinence et de performance, il ne resterait qu'à définir les réponses à assigner à chaque intention pour compléter l'assistant conversationnel. Avec la définition de notre phase d'affectation des contraintes, cette dernière étape serait par ailleurs triviale. Nous arriverions ainsi au terme de la procédure, et nous disposerions donc au final d'un assistant conversationnel fonctionnel grâce à une approche semi-supervisée dont les bases ont déjà été définies dans Schild et al. (2021).

Références

Gançarski, P. et C. Wemmert (2007). Collaborative multi-step mono-level multi-strategy classification. *Multimedia Tools and Applications 35*(1), 1–27.

Lampert, T., B. Lafabregue, et P. Gançarski (2019). Constrained Distance based K-Means Clustering for Satellite Image Time-Series. In *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, pp. 2419–2422. IEEE.

Schild, E., G. Durantin, J.-C. Lamirel, et F. Miconi (2021). Conception itérative et semi-supervisée d'assistants conversationnels par regroupement interactif des questions. *EGC 2021*.

Summary

The design of a dataset needed to train a chatbot is most often the result of manual and tedious step. To guarantee the efficiency of the annotation, we propose the interactive clustering method, an active learning method based on constraints annotation. It's an iterative approach, relying on a constrained clustering algorithm and using annotator knowledge to lead clustering. In this paper, we expose the process to design a chatbot with the interactive clustering method.

DAGOBAH : Activités de recherche Orange autour de l'annotation sémantique de données tabulaires

Yoan Chabot, Pierre Monnin

Orange, France
{yoan.chabot, pierre.monnin}@orange.com

Un grand nombre de gisements de données internes aux entreprises ainsi qu'une part non-négligeable des données du Web sont représentés sous forme de tables. La capacité à annoter ces données à l'aide de graphes de connaissances est cruciale et permet d'ouvrir la voie à de nouveaux services basés sur la sémantique (Chabot et al., 2019a).

Dans cet exposé, nous définirons le problème de l'annotation sémantique et dresserons un panorama des approches existantes. Nous structurons cet état de l'art autour d'une décomposition classique en trois étapes :

— CEA (Cell-Entity Annotation) visant à associer, à chaque cellule d'une table, une ou plusieurs entités d'un graphe de connaissances à l'aide de techniques de lookups syntaxiques, d'alignement d'ontologies ou encore de plongements de graphes (Efthymiou et al., 2017; Kilias et al., 2018).

— CTA (Column-Type Annotation) dont le but est d'associer, à chaque colonne de la table, un type issu du graphe de connaissances à l'aide de méthodes telles que le vote majoritaire (Mulwad et al., 2010).

— CPA (Columns-Property Annotation), enfin, permettant d'identifier des propriétés sémantiques entre des paires de colonnes (Ran et al., 2015).

Nous aborderons ensuite les enjeux de l'annotation de données tabulaires pour une entreprise comme Orange. Les efforts de recherche du groupe sur ce sujet, cristallisés au sein d'un projet nommé DAGOBAH (Chabot et al., 2019b, 2020; Huynh et al., 2020), seront présentés avec un focus sur des techniques de plongements de graphes de connaissances pour le typage de colonnes et la désambiguïsation des cellules. Enfin, cet exposé s'attardera sur les efforts en cours au sein de la communauté scientifique autour de ces questions par le biais du challenge ISWC SemTab (Cutrona et al., 2020; Jiménez-Ruiz et al., 2020a,b).

Références

Chabot, Y., P. Grohan, G. L. Calvez, et C. Tarnec (2019a). Dataforum : Faciliter l'échange, la découverte et la valorisation des données à l'aide de technologies sémantiques. In *Extraction et Gestion des connaissances, EGC 2019, Metz, France, January 21-25, 2019*, Volume E-35 of *RNTI*, pp. 441–444. Éditions RNTI.

Chabot, Y., T. Labbé, J. Liu, et R. Troncy (2019b). DAGOBAH : an end-to-end context-free tabular data semantic annotation system. In *Proceedings of the Semantic Web Challenge*

on Tabular Data to Knowledge Graph Matching co-located with the 18th International Semantic Web Conference, SemTab@ISWC 2019, Auckland, New Zealand, October 30, 2019, Volume 2553 of *CEUR Workshop Proceedings*, pp. 41–48. CEUR-WS.org.

Chabot, Y., T. Labbé, J. Liu, et R. Troncy (2020). DAGOBAH : Un système d'annotation sémantique de données tabulaires indépendant du contexte. In *IC 2020 : 31es Journées francophones d'Ingénierie des Connaissances (Proceedings of the 31st French Knowledge Engineering Conference), Angers, France, June 29 - July 3, 2020*, pp. 120–132.

Cutrona, V., F. Bianchi, E. Jiménez-Ruiz, et M. Palmonari (2020). Tough tables : Carefully evaluating entity linking for tabular data. In *The Semantic Web - ISWC 2020 - 19th International Semantic Web Conference, Athens, Greece, November 2-6, 2020, Proceedings, Part II*, Volume 12507 of *Lecture Notes in Computer Science*, pp. 328–343. Springer.

Efthymiou, V., O. Hassanzadeh, M. Rodriguez-Muro, et V. Christophides (2017). Matching web tables with knowledge base entities : From entity lookups to entity embeddings. In *The Semantic Web - ISWC 2017 - 16th International Semantic Web Conference, Vienna, Austria, October 21-25, 2017, Proceedings, Part I*, Volume 10587 of *Lecture Notes in Computer Science*, pp. 260–277. Springer.

Huynh, V., J. Liu, Y. Chabot, T. Labbé, P. Monnin, et R. Troncy (2020). DAGOBAH : enhanced scoring algorithms for scalable annotations of tabular data. In *Proceedings of the Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab 2020) co-located with the 19th International Semantic Web Conference (ISWC 2020), Virtual conference (originally planned to be in Athens, Greece), November 5, 2020*, Volume 2775 of *CEUR Workshop Proceedings*, pp. 27–39. CEUR-WS.org.

Jiménez-Ruiz, E., O. Hassanzadeh, V. Efthymiou, J. Chen, et K. Srinivas (2020a). Semtab 2019 : Resources to benchmark tabular data to knowledge graph matching systems. In *The Semantic Web - 17th International Conference, ESWC 2020, Heraklion, Crete, Greece, May 31-June 4, 2020, Proceedings*, Volume 12123 of *Lecture Notes in Computer Science*, pp. 514–530. Springer.

Jiménez-Ruiz, E., O. Hassanzadeh, V. Efthymiou, J. Chen, K. Srinivas, et V. Cutrona (2020b). Results of semtab 2020. In *Proceedings of the Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab 2020) co-located with the 19th International Semantic Web Conference (ISWC 2020), Virtual conference (originally planned to be in Athens, Greece), November 5, 2020*, Volume 2775 of *CEUR Workshop Proceedings*, pp. 1–8. CEUR-WS.org.

Kilias, T., A. Löser, F. A. Gers, R. Koopmanschap, Y. Zhang, et M. Kersten (2018). Idel : In-database entity linking with neural embeddings.

Mulwad, V., T. Finin, Z. Syed, et A. Joshi (2010). Using linked data to interpret tables. In *Proceedings of the First International Workshop on Consuming Linked Data, Shanghai, China, November 8, 2010*, Volume 665 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Ran, C., W. Shen, J. Wang, et X. Zhu (2015). Domain-specific knowledge base enrichment using wikipedia tables. In *2015 IEEE International Conference on Data Mining, ICDM 2015, Atlantic City, NJ, USA, November 14-17, 2015*, pp. 349–358. IEEE Computer Society.

Détection automatique des liens d'articles dans la une des journaux en ligne

Romain Perrone*, Nada Lasri*
Előd Egyed-Zsigmond**,*

*INSA de Lyon
20 avenue Albert Einstein, 69100 Villeurbanne
prenom.nom@insa-lyon.fr
**Université de Lyon, LIRIS UMR 5250 CNRS

Résumé. La détection automatisée des liens d'articles dans la *une* des journaux en ligne est un sujet très peu étudié, bien qu'il s'agisse d'une étape clé pour extraire des informations à partir d'un journal. Dans cet étude, nous présentons une nouvelle approche permettant de détecter efficacement les liens des articles présents sur un large éventail de pages web de journaux. Notre méthode détecte les liens présents sur une page web et élimine les liens non pertinents grâce à des techniques d'apprentissage automatique basée sur le texte des liens. Les attributs DOM des balises *liens* sont ensuite regroupés sous forme de clusters et une série d'expressions XPATH représentant les groupements d'articles sur la page est générée. L'algorithme produit une liste d'url d'articles en sortie. Un des avantages de notre approche est qu'elle ne repose pas sur la structure initiale du DOM : elle donne donc de très bons résultats même face à des mises en page totalement différentes.

1 Introduction

Les journaux *en ligne* permettent d'identifier des informations clés. Ils sont souvent utilisés par les industriels pour surveiller l'arrivée de nouveaux compétiteurs, dans le but d'assurer une veille concurrentielle

Pour ce faire, il est nécessaire d'obtenir dans un premier temps les liens des articles publiés par les journaux. Plusieurs techniques existent, à savoir : **1.** la détection d'articles à partir du flux RSS ; **2.** la détection d'articles à partir du site web principal du journal, dite *la une*.

L'un des inconvénients majeurs de la détection par flux RSS (Han et al., 2009) est qu'elle ne fonctionne pas sur tous les sites de journaux. Afin de pallier ces problématiques, nous avons choisi de mettre en place une méthode pour extraire les liens pointant vers les articles individuels à partir de la une des journaux. Extraire les articles de cette façon est une tâche difficile, car ces derniers se trouvent souvent mélangés à du contenu annexe comme les menus de navigation, les encarts publicitaires et les en-têtes et pieds de page. Par ailleurs, la mise en page et la structure du DOM HTML des sites web des journaux est souvent très différente d'un site à l'autre.

Dans cet article, nous proposons une nouvelle approche pour détecter automatiquement les articles à partir de la une des journaux. L'algorithme proposé ne s'appuie pas sur la mise en page du site web, et n'a pas besoin d'analyser la page avant l'extraction.

2 Travaux connexes

Peu de travaux ont été publiés au sujet de l'extraction automatisée des liens d'articles dans la une des journaux.

Dans le domaine de l'extraction d'information, le contenu est généralement récupéré en deux étapes. 1. Dans un premier temps, les liens vers les articles sont détectés grâce à un flux RSS ou à la une des journaux ; 2. Puis, pour chaque article, l'identification du titre et du contenu est réalisée à partir des URLs trouvées précedemment.

Jusqu'à présent, la quasi-majorité des recherches dans le domaine de l'extraction d'information s'est concentrée sur l'extraction du titre et du contenu de l'article **et non sur la détection des liens en amont**. Zhou et al. (2007) ont analysé les pages web sous la forme d'arbres visuels et ont extrait des descripteurs pour les différents blocs dans le but d'entraîner un algorithme d'apprentissage automatique pour arriver à un wrapper performant.

Deux approches existent pour l'extraction automatique des liens d'articles. Ces dernières reposent sur la génération d'un wrapper qui permet d'identifier la balise lien. La technique de *wrapper induction* (Kushmerick et al., 1997) utilise l'apprentissage supervisé pour apprendre des règles d'extraction à partir d'un ensemble d'exemples annotés à la main. Les descripteurs extraits sont propres au site internet : attributs d'une balise DOM, taille du texte, etc. Les inconvénients de cette technique sont : - le temps nécessaire à l'annotation des exemples à la main : il est nécessaire d'entraîner un wrapper pour chaque site internet ; et - la maintenance du wrapper : en cas de changement sur le site internet, le wrapper doit être entraîné à nouveau.

À cause de la charge de travail importante qu'engendre cette méthode, il est difficile d'extraire des informations à partir de sites internet différents, étant donné que chaque site a sa propre structure et demande une annotation manuelle pour entraîner le wrapper.

Une autre approche consiste à utiliser l'apprentissage non supervisé pour générer le wrapper. On parle d'*automated wrapper generation* (Xia et al., 2011; Jindal, 2005). L'extraction automatisée est possible car la plupart des objets web suivent des modèles fixes. La découverte de ces modèles ou schémas permet au système d'effectuer l'extraction automatiquement.

Notre méthode génère automatiquement un wrapper (*automated wrapper generation*) capable de détecter les liens d'articles. Cela nécessite une annotation manuelle des liens sur quelques sites internet par langue, pour l'entraînement d'un modèle TAL (voir section 3.1) puis une étape de clustering (voir section 3.2.3). Notre wrapper peut être utilisé sur des sites internet autres que ceux utilisés pour l'entraînement du modèle, à condition qu'ils soient dans la même langue.

Notre méthode offre : - un coût de maintenance réduit ; et - une plus grande flexibilité face à des mises en pages et des structures de pages différentes et changeantes.

3 Extraction des liens

Notre méthode se base sur deux approches complémentaires. L'une exploite les caractéristiques des textes des liens pointant vers des articles individuels, contenant en général leur titre. Nous avons mis en place un modèle qui est "capable de reconnaître" un titre parmi d'autres textes de liens. La deuxième approche se base sur des caractéristiques des éléments DOM de la page html contenant les liens. Nous nous sommes rendus compte qu'il suffit d'étudier uniquement l'élément lien (balise <a>) et son élément DOM parent et leurs attributs afin de pouvoir identifier des liens vers des articles.

3.1 Approche basée sur le contenu textuel des liens

Le problème principal en ce qui concerne le scrapping des *une* de journaux est la capacité de distinguer entre un lien pertinent qui ponte vers un article individuel et un lien non pertinent qui pointe vers des sous-rubriques, impressum, partenaires, ... Lorsque l'on parle de lien pertinent, nous définissons cela comme un lien du site de la *une* de journal qui mène vers la page web d'un article unique.

3.1.1 Classification de textes

En observant la structure des pages HTML des sites de journaux, un aspect constant émerge : un lien, l'attribut *href* d'une balise de type <*a*>, est toujours associé à un texte, une phrase la plupart du temps, qui est le titre de l'article pointé. Nous cherchons à classifier ce lien en se basant sur le texte qu'il contient.

```
<a class="css-kej3w4"
href="https://www.nytimes.com/2020/12/15/us/coronavirus-vaccine
-doses-reserved.html">
With First Dibs, Rich Countries Have 'Cleared the Shelves'
</a>
```

Ainsi, comme on peut le voir dans l'exemple ci-dessus, le lien `https://www.nytimes.com/2020/12/15/us/coronavirus-vaccine-doses-reserved.html` a pour texte associé "With First Dibs, Rich Countries Have 'Cleared the Shelves'".

A partir de ces textes, nous avons réalisé plusieurs classifications. Le but est de mettre au point un modèle capable de discerner entre un titre d'article de journal et une phrase non pertinente. Par exemple, la phrase "For the first time, Trump says he'll go if Electoral College votes for Biden" doit être interprétée comme un titre d'article, et la phrase "View your subscriber options" doit être considérée comme une phrase non pertinente.

3.1.2 Création d'un jeu de données d'apprentissage

Pour mettre au point un modèle, nous avons eu besoin de données d'apprentissage. La tâche spécifique qui anime notre projet, le scrapping de *une* de journaux, n'a pas encore été fortement explorée, nous n'avons donc pas pu trouver de jeu de données sur lequel entraîner notre modèle.

Pour pallier cela, nous avons construit nous même notre jeu d'apprentissage pour des journaux anglophones. Pour ce faire, nous avons choisi quatre sites web : *https://www.nytimes.com*, *https://www.reuters.com*, *https://www.foxnews.com*, *https://www.theguardian.com*.

À partir de chaque site, nous avons extrait tous les liens présents dans les balises <a>, et associés à du texte non vide. En récupérant ces couples (lien, texte), on insère les données dans un document csv, qui sera annoté à la main.

À l'issue de ce travail, on obtient une classification binaire : la classe des articles, annotée d'un 1 et la classe des liens non pertinents, annotée d'un 0.

Cette classification manuelle est à faire pour chaque langue. Si on souhaite extraire des liens à partir de journaux francophones ou dans d'autres langues, il faut reproduire cette phase d'entraînement. En général quelques centaines de titres de liens annotés suffisent. Ensuite le modèle sera capable de classer automatiquement les liens en se basant sur leurs contenu textuelle.

Pour améliorer la précision de nos modèles, on sépare ces liens en données d'entraînement et de validation avec un découpage 80%/20% sur le jeu d'apprentissage complet. En ce qui concerne les liens de test, on les récupérera d'un site différent des quatre mentionnés précédemment : The Economist (`https://www.economist.com`). Notre jeu de données totalise les nombres de liens précisés dans le tableau TAB. 1.

	Nombre de liens
Jeu d'entraînement	448
Jeu de validation	114
Jeu de test	77
Total	639

TAB. 1: Taille du jeu de données obtenu

Nous avons implanté deux techniques d'apprentissage pour classer les liens.

3.1.3 Modèle Naive Bayes

Pour commencer, nous avons décidé d'appliquer un algorithme supervisé sur notre jeu de données. Face au large choix d'algorithmes existants, nous avons choisi de commencer par une des méthodes traditionnelles dans l'apprentissage automatique, le Naive Bayes.

Cette méthode a pour avantage d'être un classifieur très rapide comparé à d'autres méthodes plus sophistiquées, grâce au découplage des distributions des caractéristiques conditionnelles de classe. Cela signifie que chaque distribution peut être estimée indépendamment comme une distribution unidimensionnelle.
Cela permet à son tour d'atténuer les problèmes liés au fléau de la dimension. De plus, les méthodes Naive Bayes nécessitent peu de données d'entraînement pour estimer les paramètres nécessaires.

Pour entraîner notre modèle, on s'appuie sur un sac de mots, un vecteur de comptage pour chaque phrase. On appliquera TD-IDF pour normaliser les entrées, qui seront ensuite fournies au modèle pour l'entraînement.

3.1.4 Modèle BERT

Ensuite, nous avons voulu observer les résultats d'un modèle plus lourd, mais qui possède une puissance décuplée par rapport à la méthode plus simple choisie précédemment. Pour ce faire, nous avons choisi d'étudier le classifieur BERT, publié par des chercheurs de Google AI Language (Devlin et al., 2019).

En effet, ce classifieur a des résultats de niveau de l'état de l'art dans une grande variété de tâches liées au TAL. Ce modèle nous a particulièrement intéressé, car il présente une innovation technique intéressante par rapport au reste. Il s'agit d'appliquer l'entraînement bidirectionnel de Transformer (un modèle d'attention populaire qui apprend les relations contextuelles entre les mots) à la modélisation du langage.
A partir d'un modèle BERT pré-entrainé, nous entraînons notre modèle BERT personnalisé grâce à notre jeu de donnés annoté décrit précédemment.

La taille des textes que nous traitons est de l'ordre de quelques mots, nous avons donc décidé de choisir un modèle de type BertForSequenceClassification (Sanh et al., 2019).
Vu la taille des données dont nous disposons, nous avons décidé de travailler en mode distilled avec le modèle TFDistilBertForSequenceClassification, qui est plus rapide et léger que celui mentionné précédemment. Deux classes seront retenues donc deux labels (0 et 1). L'optimisation est réalisée avec l'optimiseur Adam et un learning rate de 5e-5. L'entrainement est réalisé avec un batch de taille 16 et trois epoch. On obtient, pour le jeu de donnés d'entraînement et de validation les résultats du tableau TAB. 2.

Le classifieur basé sur BERT nous produit des prédictions sous la forme de valeur entre 0 et 1. Après quelques essais, nous considérons qu'il faut atteindre un score de 0.9 pour être considéré comme un article.

	train accuracy	val accuracy
Epoch 1	0.829	0.920
Epoch 2	0.964	0.947
Epoch 3	0.976	0.947

TAB. 2: Résultats intermédiaires du classifieur BERT lors de l'entrainement

On observe de très bons scores au training, avec déjà plus 0.98 de précision au bout de trois epoch. De plus, on remarque que la précision du jeu de donnés de validation est peu éloignée de la précision du jeu de donnés d'entraînement (0.984 et 0.947), ce qui renforce l'efficacité de notre modèle.

On décide donc de comparer ces deux modèles, d'un côté pour observer leurs comportements et leurs efficacités face à un nouveau site web, mais aussi pour observer à quel point l'utilisation d'un modèle lourd va améliorer la précision du modèle par rapport à la méthode classique.

3.1.5 Résultats sur le fichier test

Nous souhaitons observer les résultats de nos deux algorithmes, et pour ce faire nous utiliserons le fichier de test présenté en 3.1.2.

	Précision	Rappel	F1
Naive Bayes	0.690	0.808	0.744
BERT	0.913	0.808	0.857

TAB. 3: Comparaison des résultats obtenus pour les méthodes Naive Bayes et BERT sur le jeu de données de test

Sur les chiffres du tableau TAB. 3 on remarque que les performances de précision du modèle Naive Bayes sont inférieures que celles du modèle basé sur BERT, avec des valeurs qui passent de 0.690 à 0.913. En effet, le modèle basé sur Naive Bayes va détecter 29 liens articles, mais dont seulement 21 d'entre eux sont réellement des liens d'articles. De son côté, BERT va en récupérer seulement 23, avec 21 vrais liens d'articles.

En ce qui concerne le rappel, il est le même pour les deux méthodes, sur les 26 liens à récupérer, les deux méthodes vont rater 5 de ces liens.

3.1.6 Règle explicite

Nous avons observé que dans la majorité des cas, les titres d'articles sont composés de 4 mots ou plus. Les liens non pertinents, au contraire, sont la plupart du temps associé à un texte succinct, avec des impératifs (Subscribe / Read full edition) et parfois seulement un mot, le nom d'une section. Pour améliorer la performance de notre algorithme, nous avons donc mis en place la condition suivante : **tout lien associé à un titre de moins de 4 mots est directement considéré comme non pertinent**, quelle que soit la prédiction réalisée par le modèle de classification. On peut observer sur le tableau TAB. 4 que pour la méthode de Naive

	Précision	Rappel	F1
Naive Bayes	0.690	0.808	0.744
Naive Bayes + règle	0.944	0.692	0.799
BERT	0.913	0.808	0.857
BERT + règle	1.000	0.731	0.844

TAB. 4: Comparaison des résultats obtenus pour les méthodes Naive Bayes et BERT et la règle explicite sur le jeu de données de test

Bayes, la performance du classifieur est améliorée, passant de 0.805 à 0.870, ce qui est notable. Dans le cas du classifier basé sur BERT, on atteint une précision de 1.0, mais au détriment de la performance du rappel qui diminue de 0.808 à 0.731. Le Rappel est aussi affecté en ce qui concerne le modèle de Naive Bayes.
Cete règle nous fait perdre certains liens d'article scrapés, mais globalement, elle améliore les résultats.

3.2 Approche basée sur les attributs DOM

En complément de l'approche basée sur le contenu textuel des liens, nous avons implémenté une seconde approche qui exploite les attributs des éléments de la structure du DOM HTML. On identifie dans l'arbre HTML les balises <*a*> contenant les liens extraits en 3.1., puis on extrait des caractéristiques DOM pour un apprentissage non supervisé (clustering). On génère en sortie une liste d'expressions XPATH permettant de récupérer les liens d'article.

3.2.1 Prétraitement

Pour préparer l'étape de classification, il est nécessaire d'extraire des caractéristiques au niveau de l'arbre DOM de la page web. Ces descripteurs s'apparentent à un ensemble de couples (attribut, valeur), où *attribut* peut être de type : class, data, data-*, id ou avoir une valeur spéciale définie par l'utilisateur.

```
<a class="css-kej3w4"
href="https://www.nytimes.com/2020/12/15/us/coronavirus-vaccine
-doses-reserved.html"
data-uri="nyt://article/caecb59e-9fde-5ba0-9c24-8a85680e14e8"
data-story="nyt://article/caecb59e-9fde-5ba0-9c24-8a85680e14e8"
data-visited="">
</a>
```

Dans l'exemple suivant, on a :

balise	attribut	valeur
a	class	css-kej3w4
a	data-uri	nyt://article/caecb59e-9fde-5ba0-9c24-8a85680e14e8
a	data-story	nyt://article/caecb59e-9fde-5ba0-9c24-8a85680e14e8
a	href	https://www.nytimes.com/2020/12/15/us/coronavirus-vaccine-doses-reserved.html

Pour rappel, chaque élément HTML peut avoir un ou plusieurs attributs. L'attribut *class* par exemple, permet de définir la ou les classes auxquelles appartient un élément afin le mettre en forme avec une feuilles de style CSS.

Nous allons réaliser un apprentissage supervisé à partir des attributs des éléments <*a*> et de leur parent (div, span, li, ...). Tous ces attributs sont récupérés et ajoutés à la structure de données de la table TAB. 5 que nous appellerons tableau *preprocessing*.

La colonne *nombre* comptabilise le nombre de fois que le couple (attribut,valeur) apparaît au niveau de la balise <*a*> ou de son parent. Cette valeur est essentielle, car elle permet de supprimer les attributs inutiles. Ce processus de suppression est détaillé dans la section 3.2.2. Dans cet exemple la balise : a été rencontrée 19 fois.

Comme l'attribut $href$ est unique pour chaque article, il ne permet pas d'identifier des clusters. Nous ne le prendrons pas en compte dans nos traitements.

3.2.2 Suppression des attributs inutiles

Nous recherchons des articles présentant des attributs similaires (ex. class="headline-link"), c'est à dire des couples (attribut, valeur) présents plusieurs fois dans le document. Si un couple

attribut	valeur	nombre
class	headline-link	19
class	ds-link-with-arrow	4
class	ds-link-with-arrow–minor	3
class	current-edition__flashes-item	3
data-analytics	graphic_detail :headline	1
data-analytics	economist_today :headline_5	1
data-analytics	weekly_edition :flash_2	1
data-analytics	economist_today :headline_7	1
data-analytics	economist_today :headline_3	1

TAB. 5: Exemple de prétraitement réalisé à partir de la une de The Economist.

(attribut, valeur) n'est présent qu'une seule fois, nous pouvons l'éliminer. Pour chaque élément <a> du DOM : pour chaque couple *(attribut, valeur)* de l'élément et de son parent : on parcourt *preprocessing* à la recherche de cette valeur, en s'assurant que l'attribut correspond. Si le compteur est <= 1, l'attribut est unique. Il peut être supprimé de la liste des attributs à considérer. Mais si la valeur contient un ou plusieurs nombres (ex. `economist_today:headline_3`), on parcourt à nouveau *preprocessing* en remplaçant les chiffres par l'expression regex « [0-9]+ ». En effet, les balises
`<a data-analytics="economist_today:headline_3">` et
`<a data-analytics="economist_today:headline_7">` peuvent être considérées comme similaires au chiffre près. Si le compteur est toujours <= 1 on supprime l'attribut de la liste des attributs à considérer.

Cette étape permet de se débarrasser des couples *(attribut, valeur)* n'ayant pas une portée générale.

3.2.3 Clustering

A partir des couples (attribut, valeur) retenus pour chaque élément <a> du DOM, on cherche à identifier des clusters.

On décide de créer deux vecteurs (v_a et v_{parent}) : pour représenter les deux niveaux de l'arborescence du DOM. Nous aurions pu considérer l'ensemble des couples (attribut,valeur) de l'arborescence DOM jusqu'à la balise <a>, mais en pratique les attributs de <a> et de son parent suffisent.

Chaque vecteur peut s'exprimer comme une liste finie d'éléments formatés ainsi :
$balise.niveau.attribut = valeur$ avec $balise$ = p, div, a, h (équivalent à h1, h2, h3, h4, h5 et h6) ; et $niveau$ = 0 pour la balise <a> et 1 pour la balise parent.

Par exemple pour une balise parent <h2 name="test" class="hello"></h2>, on obtiendra le vecteur $v_{(texte)parent}$ suivant : (h.1.name=test, h.1.class=hello)

Pour obtenir un vecteur au format numérique, nous générons un sac de mots à partir de l'ensemble des $balise.niveau.attribut = valeur$ générées.

```
{
a.0.class=ds-link-with-arrow--minor,
```

```
a.0.class=ds-link-with-arrow,
a.0.data-analytics=in_context:headline_[0-9]+,
a.0.class=headline-link,
a.0.data-analytics=us_[0-9]+_election:headline_[0-9]+,
a.0.data-analytics=weekly_edition:flash_[0-9]+,
a.0.data-analytics=economist_today:headline_[0-9]+,
li.1.class=current-edition__flashes-item,
a.0.data-analytics=readers_favourites:headline_[0-9]+
}
```

On exprime alors le vecteur v en encodage one-hot, à partir de v_a et v_{parent}. La balise

```
<li class="current-edition__flashes-item">
    <a href="..." class="headline-link"> </a>
<li>
```

sera codée sous la forme :

```
(0,0,0,1,0,0,0,1,0)
```

On peut maintenant chercher à identifier des clusters à partir des vecteurs v générés.

Le nombre de clusters étant inconnu à l'avance on utilisera la méthode DBScan avec $eps = 0.5$, $min\_samples = 2$, obtenus expérimentalement. La similarité cosinus étant fréquemment utilisée pour mesurer la similarité entre deux vecteurs, nous l'utiliserons comme métrique de distance.

	cluster
['a.0.data-analytics=readers_favourites:headline_[0-9]+', 'a.0.data-analytics=economist_today:headline_[0-9]+', 'a.0.class=headline-link', 'a.0.data-analytics=us_[0-9]+_election:headline_[0-9]+', 'a.0.data-analytics=in_context:headline_[0-9]+']	1
['a.0.data-analytics=weekly_edition:flash_[0-9]+', 'li.1.class=current-edition__flashes-item', 'a.0.class=ds-link-with-arrow–minor', 'a.0.class=ds-link-with-arrow']	2

TAB. 6: Clusters identifiés à partir de la une de The Economist.

Le tableau présenté en TAB. 6 montre les balises, attributs, valeurs apparaissant au moins une fois dans un cluster donné.

3.2.4 Simplification du patron

Une fois les clusters identifiés, on cherche à réduire la variance intra-groupe. Pour ce faire, on fixe un *threshold = 0.75*. Pour chaque dimension, on calcule la valeur moyenne des vecteurs du cluster pour cette dimension. Si cette valeur moyenne est $> threshold$ on conserve les

descripteurs associées à cette dimension. Sinon, on simplifie le patron (pattern) en ignorant ces descripteurs afin de réduire la variance intra-groupe.

Le tableau présenté en TAB. 7 correspond aux balises, attributs, valeurs retenues après simplification. Le tableau contient moins d'attributs que celui présenté en TAB. 6 : c'est exactement ce que l'on recherchait.

3.2.5 Génération du XPATH

Pour finir, on génère une expression XPATH à partir du patron simplifié comme illustré dans la colonne XPATH du tableau TAB. 7. Les expressions XPATH générées permettent de récupérer les liens d'articles sur la page. Nous utilisons la fonction *contains*, car les attributs peuvent avoir des valeurs composées.

attribut	XPATH	cluster
['a.0.class=headline-link']	//a[contains(@class, 'headline-link')]	1
['a.0.class=ds-link-with-arrow']	//a[contains(@class, 'ds-link-with-arrow')]	2

TAB. 7: Clusters simplifiés (The Economist)

4 Résultats

Nous souhaitons comparer les performances de 3 méthodes d'extraction des liens : méthode basée sur le texte (Text based) (3.1.4), méthode basée sur les attributs DOM (DOM) (3.2.3), méthode basée sur la combinaison des deux méthodes (Text + DOM) .
Pour cela, nous avons choisi d'extraire les unes de quatre sites de journaux :

— Le New York Times : `https://www.nytimes.com`
— The Economist : `https://www.economist.com`
— The Financial Times : `https://www.ft.com`
— The Times UK `https://www.thetimes.co.uk`

Afin de garantir des résultats fiables, toutes les méthodes décrites ici simulent l'interaction du site web avec un navigateur web (le défilement jusqu'en bas de la page), afin de pouvoir récupérer la totalité des articles chargés dynamiquement (*lazy-loading*).

Text based. L'ensemble des balises <a> présentes sur la page sont récupérées (à l'exclusion de celles présentes dans le header et le footer). Il s'agit ensuite de réaliser des prédictions : lien vers un article ou non, à partir du texte contenu dans ces balises (3.1.4)

DOM. L'ensemble des balises <a> présentes sur la page sont récupérées (à l'exclusion de celles présentes dans le header et le footer). Il s'agit ensuite d'extraire des attributs DOM pour effectuer un apprentissage non supervisé (clustering) et détecter les liens vers les articles (3.2.3).

Text + DOM. Les deux approches (Text based et DOM) sont combinées.

L'ensemble des balises <a> présentes sur la page sont récupérées (à l'exclusion de celles présentes dans le header et le footer). Puis tous les liens détectés par la méthode Text based sont fournis à la méthode DOM, qui récupère à son tour les liens vers les articles après un

apprentissage non supervisé (clustering). On observe ainsi que quelle que soit le site, c'est

	Précision FT	Rappel FT	Précision Times	Rappel Times
Text based	0.956	**1.0**	0.906	**0.943**
DOM	0.956	0.93	**0.926**	0.927
Text + DOM	**0.985**	0.92	0.915	0.926

TAB. 8: Résultats pour le Financial Times (FT) et le Times UK (Times)

la combinaison des deux modèles qui permet d'obtenir la meilleure précision. Dans le cas du New York Times, les deux algorithmes ont des précisions proches (0.956 pour l'approche basée sur le contenu textuel des liens et 0.961 pour la méthode basée sur la structure de l'arbre HTML), et leur combinaison a la meilleure précision (0.985). Dans le cas de The Economist, les deux algorithmes ont une précision de 1, ce qui fait que leur combinaison aussi. Dans le cas du Financial Times, la méthode basée sur la structure de l'arbre HTML a pu être combinée avec l'approche basée sur le contenu textuel des liens, pour produire un résultat meilleur que les deux approches prises séparément (précision de 0.985 contre 0.956 et 0.956). Enfin dans le cas du Times UK, l'approche basée sur le DOM a une précision légèrement plus élevée (0.926) que l'approche basée sur le texte (0.906) et que les deux approches réunies (0.915). En ce qui concerne le Rappel, on observe que globalement c'est l'approche basée sur le contenu textuel des liens qui obtient les meilleurs Rappels (1.0, 0.785, 1.0 et 0.943 pour le New York Time, The Economist, le Financial Times et le Times UK). La méthode basée sur la structure de l'arbre HTML obtient des Rappels légèrement inférieurs (0.922, 0.714 pour les deux premiers sites), et cela entraîne une baisse du Rappel de la méthode combinée. Cette perte en Rappel est compensée par une meilleure précision, à part dans le cas de The Economist où la précision était déjà optimale, à 1. Enfin pour The Economist, la valeur du Rappel est globalement plus faible (0.785 contre 0.92 pour les autres sites) car 5 articles sur 28 (17%) ont un titre de moins de 4 mots. Comme expliqué dans la section Règle explicite (3.1.6) nous avons fait le choix de catégoriser tous les liens ayant un titre de moins de 4 mots comme n'étant pas des articles, et ce afin d'augmenter la précision de notre modèle. Si cette règle donne un Rappel inférieur pour The Economist, il s'agit très certainement d'une exception. En effet, les scores de rappel du New York Times (1.0), du Financial Times (1.0) et du Times UK (0.94) ne semblent pas être affectés par cette règle.

5 Conclusion et perspectives

Dans cet article, nous avons mis en oeuvre un système d'extraction automatique de liens vers des articles individuels, combinant deux approches de classification de liens d'articles dans la *une* de journaux. Nous avons évalué les performances de ces modèles sur des vrais sites web, pour observer leurs capacités à discerner un lien d'article d'un lien non pertinent. Finalement, l'implémentation de nos modèles représente une manière automatisée d'extraire des données pertinentes des sites de *une* de journaux, pour ensuite les exploiter de diverses manières. Notre méthode a besoin d'être entraînée séparément pour chaque langue. Nous l'avons

fait pour l'anglais. Notre modèle est capable désormais de fonctionner automatiquement pour cette langue. Il est tout à fait possible d'entraîner le modèle pour d'autres langues également.

Pour la suite, nous prévoyons d'étudier l'extraction du titre et du texte d'un article, dont le lien a été récupéré grâce à nos méthodes, pour exploiter les données de la page de l'article. Il aurait aussi été intéressant d'étudier la validité de notre méthode sur un temps long, pour observer l'évolution du scrapping et des performances sur des plus de données dans le temps. Nous allons également améliorer notre méthode afin qu'il extraie non seulement les liens des articles à partir de la *une* d'un journal en ligne, mais également leurs catégories quand elles sont précisées. C'est une information qui ne figure pas dans les flux RSS, mais est possible d'extraire à partir de la page principale du journal, voir parfois, à partir des pages d'articles individuels. Enfin, nous envisageons de généraliser la méthode sur plusieurs langues, comme pour le français, l'allemand... Il suffit de réaliser le training du modèle avec la langue souhaitée.

Références

Devlin, J., M.-W. Chang, K. Lee, et K. Toutanova (2019). Bert : Pre-training of deep bidirectional transformers for language understanding. In J. Burstein, C. Doran, et T. Solorio (Eds.), *NAACL-HLT (1)*, pp. 4171–4186. Association for Computational Linguistics.

Han, H., T. Noro, et T. Tokuda (2009). An automatic web news article contents extraction system based on rss feeds. *J. Web Eng. 8*(3), 268–284.

Jindal, N. (2005). Wrapper generation for automatic data extraction from large web sites. In S. Bhalla (Ed.), *Databases in Networked Information Systems*, Berlin, Heidelberg, pp. 34–53. Springer Berlin Heidelberg.

Kushmerick, N., D. S. Weld, et R. B. Doorenbos (1997). Wrapper induction for information extraction. *IJCAI*.

Sanh, V., L. Debut, J. Chaumond, et T. Wolf (2019). Distilbert, a distilled version of BERT : smaller, faster, cheaper and lighter. *CoRR abs/1910.01108*.

Xia, Y., Y. Yang, S. Zhang, et H. Yu (2011). Automatic wrapper generation and maintenance. In *Proceedings of the 25th Pacific Asia Conference on Language, Information and Computation*, Singapore, pp. 90–99. Institute of Digital Enhancement of Cognitive Processing, Waseda University.

Zhou, D., C. Giles, S. Zheng, et J. Li (2007). Extracting author meta-data from web using visual features. In *2007 7th IEEE International Conference on Data Mining Workshops*, Los Alamitos, CA, USA, pp. 33–40. IEEE Computer Society.

Comparaison de méthodes d'extraction de mots-clés non supervisées

Alaric Tabariès[1] et David Reymond[2]

IMSIC, Université de Toulon, Toulon, France
alaric-tabaries@etud.univ-tln.fr
IMSIC, Université de Toulon, Toulon, France
david.reymond@univ-tln.fr

Résumé Avec l'émergence de l'accès libre et gratuit aux données scientifiques, la volumétrie d'informations accessibles par le chercheur augmente de manière exponentielle. Cette nouvelle dynamique informationnelle rend le processus de veille documentaire, essentiel à la recherche scientifique, tant complexe que chronophage. C'est dans ce contexte que l'extraction d'information se pose en tant que service support au prétraitement de la sélection documentaire. En effet, les mots-clés, qui représentent les sujets principaux traités dans un document, sont particulièrement utiles pour distinguer les ressources intéressantes dans un ensemble imposant de documents. Cependant, très peu en sont pourvus. L'extraction automatique de mots-clés permet de remédier à ce problème et montre d'ores et déjà des résultats satisfaisants sur des corpus de référence. Il a cependant été établi que les disciplines scientifiques dont relèvent les documents influent sur les performances des méthodes d'extractions. Dans cet article, nous ciblons en premier lieu le degré qualitatif du résumé et sa suffisance pour avoir recours à des méthodes extractives en vue de mettre à disposition d'une communauté scientifique variée un outil d'extraction automatique adapté.

Mots-clés: Recherche d'information · Extraction de l'information · Mots-clés · Source d'information

1 Introduction

Avec l'émergence de l'accès libre et gratuit aux données scientifiques, la volumétrie d'informations accessibles par le chercheur augmente de manière exponentielle : à ce jour, plus de 1,8 millions de documents scientifiques sont accessibles sur la plateforme d'archivage internationale arXiv soit près de 12% de plus qu'en 2019. Des initiatives françaises existent également : les plateformes HAL et ISTEX connaissent un essor similaire avec, respectivement, 2,5 et 23 millions de documents référencés. C'est dans ce contexte que l'extraction d'information se pose en tant que service support au pré-traitement de la sélection documentaire. En effet, les mots-clés, qui représentent les sujets principaux traités dans un document, sont particulièrement utiles pour distinguer les ressources intéressantes. Cependant, très peu en sont pourvus : nous avons mesuré qu'environ 30% des références sur la plateforme d'archivage HAL (déc. 2020) en possèdent. L'extraction automatique de mots-clés, permet de générer des mots mots-clés issus de l'auteur même du texte, et en ce sens est moins subjective qu'un lecteur/annotateur. Les différentes techniques extractives montrent d'ores et déjà des résultats satisfaisants sur des corpus de référence (Hasan & Ng, 2014). Il a cependant été établi que les disciplines scientifiques dont relèvent les documents influent sur les performances des méthodes d'extractions (Bougouin et al., 2014). Dans cet article, nous ciblons en premier lieu le degré qualitatif du résumé et sa suffisance pour avoir recours à des méthodes extractives en vue de mettre à disposition d'une communauté scientifique variée un outil d'extraction automatique adapté. Ainsi, nous nous interrogeons sur la fréquence d'apparition de mots-clés auteur dans les résumés des articles.

2 Expérimentation

2.1 Contexte

En préfiguration à cette expérimentation, nous avons réalisé une expérimentation sur un jeu de données d'une centaine de documents relevant des disciplines des sciences humaines et sociales dont l'objectif était de comparer les performances de différentes méthodes d'extraction non supervisées (Tabariès, 2020). À l'aide d'un jeu de données élaboré par des étudiants en linguistique qui extrayaient de manière manuelle des mots-clés des résumés pour qualifier un ensemble d'articles, nous avons comparé les résultats avec des outils d'extraction automatique. Nous avons constaté que le degré de recouvrement (la propension à extraire des mots comme l'ont fait les étudiants) des méthodes variait selon les disciplines étudiées, toutefois, le corpus de documents n'était pas assez conséquent et la qualification externe par les béotiens au regard du résumé ne nous permettent pas de poser de véritables conclusions.

Nous présentons ci-après le début d'une continuité de cette expérimentation portant sur un échantillon plus important de références HAL dont les mots-clés sont ceux des auteurs des documents.

2.2 Méthode

Nous interrogeons la plateforme HAL afin de collecter les articles annotés par, à la fois, un résumé et des mots-clés. Nous extrayons par la suite le jeu de données, composé de 4 500 articles scientifiques marqués par les mots-clés auteur, à l'aide d'un échantillonage aléatoire simple sur la collection d'articles constituée. Nous constituons également un second échantillon comportant 12 500 articles caractérisés par, à la fois, un résumé et des mots-clés et dont le texte complet est renseigné.

En moyenne, dans ces jeux de données, un article est caractérisé par 5 mots-clés, dont 2 composés, ainsi qu'un résumé de 120 mots.

Nous réalisons ensuite un traitement qui consiste à nettoyer les données textuelles en supprimant les mots vides avant d'effectuer une lemmatisation des termes à l'aide du lemmatiseur intégré à la librairie Spacy, paramétré selon le modèle *fr_core_news_sm*. Il est important de noter que nous décomposons les mots-clés composés, à titre d'exemple, le mot-clé composé "*analyse de l'image*" sera donc considéré comme deux mots-clés "*analyse*" et "*image*". Nous étudions alors la relation de présence des mots-clés renseignés par l'auteur dans le résumé d'un document. La relation de présence est analysée selon un premier angle d'étude, que l'on peut qualifier de binaire, qui consiste à définir si un mot est présent ou non. Nous analysons ensuite la similarité sémantique entre ces métadonnées en se basant sur le modèle français de wordnet (Sagot, 2017).

2.3 Résultats (en cours)

Sur le premier échantillon étudié, on constate qu'en moyenne, 38% des mots-clés sont présents dans le résumé d'un document. On constate également une disparité tant au niveau du type de document qu'au niveau de la discipline étudiée. Enfin, l'étude de la similarité sémantique telle que nous l'avons appliquée n'étend que très peu les résultats obtenus et nous paraît, par conséquent, peu pertinente à ce stade.

La figure 1 présente les taux de présence des mots-clés dans les résumés obtenus selon les différents types de documents (dans l'ordre, articles, thèses, commentaires, et mémoires) dans le premier échantillon. On constate que les mots-clés annotés aux articles sont moins présents (0.31%) dans le résumé que pour d'autres types de production.

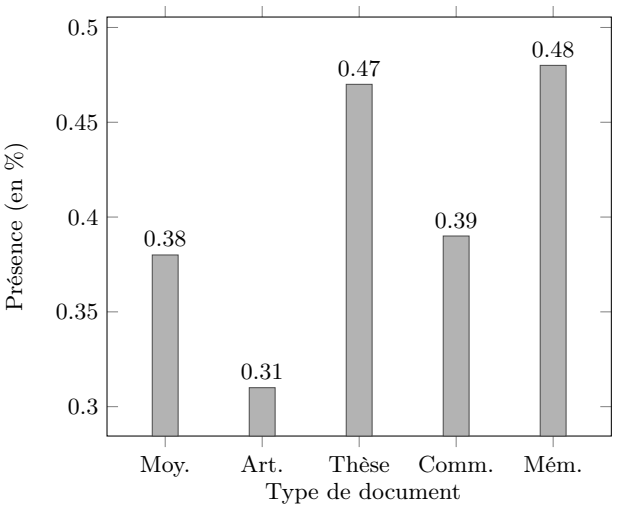


FIGURE 1: Taux de présence des mots-clés dans les résumés en fonction du type de document dans le premier échantillon

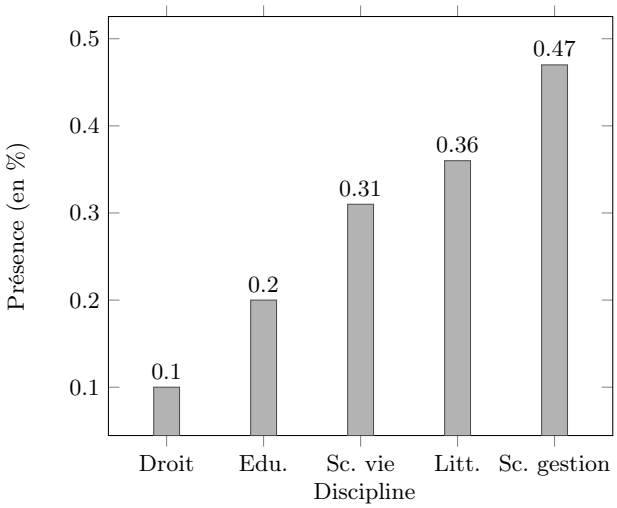


FIGURE 2: Taux de présence des mots-clés dans les résumés d'articles en fonction de la discipline dans le premier échantillon

La figure 2 présente le taux de présence des mots-clés dans les résumés obtenus selon différentes disciplines dans le premier échantillon. On constate un écart très important entre les disciplines étudiées. Pour des raisons pratiques, nous affichons les extrêmes suffisamment représentés dans l'échantillon pour illustrer le propos.

En étudiant le second échantillon, on constate que, en moyenne, 54% des mots-clés annotés par les auteurs sont présents dans les résumés des articles contre 83% dans le texte complet des articles.

3 Conclusion

La première partie de cette expérimentation, c'est-à-dire l'étude de la relation de présence des mots-clés renseignés par l'auteur dans le résumé d'un document est indispensable afin de déterminer la pertinence du résumé seul comme source de l'extraction de mots-clés et établir un niveau de référence en cas d'extension au texte plein. Les premiers résultats montrent qu'une part significative (38 % pour un échantillon généraliste, 54% pour un échantillon plus spécifique) des mots-clés auteur, bien que disparate selon les disciplines, est présente dans le résumé. Les résultats obtenus sont en accord avec le récent article de Lu et al. (2020) dans lequel les auteurs retrouvent sur une étude empirique près de 57% des mots-clés auteur dans le titre et le résumé. De plus, l'écart notable dans le taux de présence entre les différents échantillons étudiés tend à montrer qu'un nombre important de documents sont peu (voire mal) renseignés par le chercheur ce qui souligne en conséquent l'importance de l'accompagnement des chercheurs à la science ouverte au travers d'outils élaborés en ce sens (Reymond & Galliano, 2019). Il est cependant important de poursuivre cette expérimentation en définissant les modalités d'extraction les plus pertinentes selon les disciplines. En extension, nous tenterons alors de rapprocher les termes automatiquement extraits à des descripteurs issus de vocabulaires contrôlés pour tenter d'apposer une dimension qualitative supplémentaire à cette automatisation de procédés documentaires.

Références

Bougouin, A., Boudin, F., & Daille, B. (2014, juillet). Influence des domaines de spécialité dans l'extraction de termes-clés. In *Traitement Automatique des Langues Naturelles (TALN)* (pp. 13–24). Marseille, France.

Hasan, K. S., & Ng, V. (2014, juin). Automatic Keyphrase Extraction : A Survey of the State of the Art. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)* (pp. 1262–1273). Baltimore, Maryland : Association for Computational Linguistics. doi: https://doi.org/10.3115/v1/P14-1119

Lu, W., Liu, Z., Huang, Y., Bu, Y., Li, X., & Cheng, Q. (2020, novembre). How do authors select keywords ? A preliminary study of author keyword selection behavior. *Journal of Informetrics*, *14*(4), 101066. Consulté

le 2021-01-15, sur http://www.sciencedirect.com/science/article/pii/S1751157720300134 doi: https://doi.org/10.1016/j.joi.2020.101066

Reymond, D., & Galliano, C. (2019, novembre). *Cartographie de l'expertise des chercheurs de l'Université de Toulon* (Intern report). Université de toulon. Consulté le 2021-01-21, sur https://hal.archives-ouvertes.fr/hal-02643329

Sagot, B. (2017, juin). Représentation de l'information sémantique lexicale : le modèle wordnet et son application au français. *Revue française de linguistique appliquee*, *Vol. XXII*(1), 131–146. Consulté le 2020-12-17, sur https://www.cairn.info/revue-francaise-de-linguistique-appliquee-2017-1-page-131.htm?contenu=resume (Publisher : Publications linguistiques)

Tabariès, A. (2020). Comparaison de méthodes d'extraction de mots-clés non supervisées pour les disciplines des sciences humaines et sociales. In *Communications des apprenti-e-s chercheur-euse-s 2020* (Vol. 7, p. 15). Consulté sur https://jep-taln2020.loria.fr/articles-acceptes/la-conference-sur-hal/

Index