



HAL
open science

Détection automatique de similarités dans le cadre de la protection par Marque

Astrid Jourdan, Yannick Le Nir, Nicolas Girardin

► **To cite this version:**

Astrid Jourdan, Yannick Le Nir, Nicolas Girardin. Détection automatique de similarités dans le cadre de la protection par Marque. 10ème Conférence Nationale sur les Applications Pratiques de l'Intelligence Artificielle, AFIA-Association Française pour l'Intelligence Artificielle, Jul 2024, La Rochelle, France. p. 57. <hal-04639103>

HAL Id: hal-04639103

<https://hal.science/hal-04639103v1>

Submitted on 8 Jul 2024

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

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



Distributed under a Creative Commons CC BY-NC 4.0 - Attribution - Non-commercial use - International License

Détection automatique de similarités dans le cadre de la protection par Marque

A. Jourdan¹, Y. Le Nir¹, N. Girardin²

¹ CYU Cergy Paris University, ETIS

² LIPSTIP

astrid.jourdan@cyu.fr, yannick.le-nir@cyu.fr, ngirardin@lipstip.com

Résumé

Une Marque permet de protéger un signe pour des produits et services décrits sous la forme de mots ou de groupe de mots appelés items. Lorsqu'un titulaire de marque identifie une marque similaire à sa marque enregistrée, il peut former une opposition. Les produits et services de chaque marque étant décrits par des centaines d'items, l'opposition nécessite une analyse manuelle complexe de la similarité entre produits et services des deux marques. L'objectif est de proposer une méthode automatique pour trouver les similarités entre marques en utilisant les embeddings des LLMs sur les items. La mesure de similarité cosinus est ensuite intégrée avec d'autres features dans un processus d'apprentissage supervisé pour prédire la similitude. Un historique de plus de 120 000 décisions des Offices de Marques Européens permet d'évaluer la qualité des résultats

Mots-clés

Propriété Intellectuelle, Marques, Embeddings, Similarités, Apprentissage supervisé

Abstract

A trademark protects a sign for goods and services, which are described in words or groups of words called items. When a trademark owner identifies a mark similar to his registered trademark, he can file an opposition. Since the goods and services of each trademark are described by hundreds of items, opposition requires a complex manual analysis of the similarity between the goods and services of the two trademarks. The aim is to propose an automatic method for finding similarities between trademarks using LLM embeddings on items. The cosinus similarity measure is then integrated with other features in a supervised learning process to predict similarity. A history of over 120,000 European Trademark Office decisions is used to assess the quality of the results

Keywords

Intellectual property, Trademarks, Embeddings, Similarities, Supervised learning

1 Introduction

La Marque est un titre de propriété industrielle permettant une fois enregistré de protéger un signe (par exemple un nom, un logo, ...) pour des produits et services spécifiés dans une liste d'items (mots ou groupe de mots). Par exemple, la marque enregistrée SO'NUM de Digital Aquitaine protège le signe SO'NUM pour une utilisation dans environ 50 produits et services comme : Matériel informatique; logiciels; Produits de l'imprimerie; Conseil et accompagnement en matière de développement de l'économie numérique et de la digitalisation des entreprises.

Une opposition de marque vise à démontrer qu'une marque nouvellement créée porte atteinte à une ancienne marque. Pour cela, une partie essentielle vise à démontrer dans quelle mesure deux marques sont similaires ou susceptibles de créer une confusion chez les consommateurs. Cela implique notamment une analyse complexe du chevauchement potentiel en termes de produits et services. Par exemple, il faut déterminer en quoi l'item "appareils et instruments photographiques" est similaire à l'item "appareils pour la transmission du son" alors que l'item "stylos à bille" est différent de l'item "billes de stylo". Une opposition de marque peut nécessiter la comparaison de centaines de produits et services, sur la base de critères d'évaluation et de la jurisprudence (les décisions passées des offices d'enregistrement ou tribunaux). Par exemple, la comparaison de deux marques protégeant chacune 50 produits et services nécessite l'évaluation de 2500 couples d'items. Les praticiens spécialisés dans l'opposition de marque ont besoin d'outils spécifiques pour comparer rapidement des produits et services.

Il existe de nombreuses études utilisant l'intelligence artificielle pour étudier la similitude entre marques. La plupart concerne la comparaison des logo ([9], [1], [8], [13], [2]) ou des noms d'un point de vue syntaxique, sémantique ou phonétique, ou une combinaison des deux ([11], [10], [3], [12]).

A notre connaissance seuls Shmatkov et al. [7] se sont intéressés à la comparaison des produits et des services. Leur approche consiste à vectoriser les items puis utiliser une mesure de similarité cosinus ou distance de Tanimoto. Les modèles de vectorisation (*embeddings*) utilisés

sont spécifiques à la vectorisation de mots et n’intègrent que très peu le contexte de la phrase. La comparaison des items repose entièrement sur la mesure de similarité calculée et n’intègre aucune autre information telle que l’office d’enregistrement ou l’origine de la décision. Leur étude porte exclusivement sur la classe 45 de la classification de Nice. Cette classification a été instituée par l’Arrangement de Nice en 1957, et une nouvelle édition est publiée tous les cinq ans. Il s’agit d’une classification internationale de produits et de services aux fins de l’enregistrement des marques. Elle comprend actuellement 45 catégories.

Dans ce travail, nous proposons d’utiliser des *embeddings* spécifiques aux phrases grâce aux récents modèles basés sur les *Transformers* [6], puis d’intégrer la mesure de similarité calculée dans un modèle d’apprentissage de classification avec d’autres caractéristiques sur les décisions dans le but de déterminer la similarité ou dissimilarité des items pour toutes les catégories de la classification de Nice. La section 2 présente le jeu de données utilisé. Dans la section 3, nous comparons les différents modèles de *sentence embeddings* pour en extraire des similarités sémantiques des items décrivant les services ou produits. La section 4 décrit le processus d’apprentissage automatique et compare les performances des modèles de classification utilisés. Une analyse des résultats est effectuée dans la section 5, puis nous terminons avec quelques perspectives pour des travaux futurs.

2 Présentation et pré-traitement des données

L’office de marques européen EUIPO met à disposition des données concernant des pratiques ou décisions sur des paires d’items prises par différents offices européens (<https://euipo.europa.eu>) sur la période 2012 à 2023. Le jeu de données constitué regroupe 107570 pratiques et décisions renseignées par :

- un couple d’items décrivant les produits ou les services,
- la catégorie de la classification de Nice de chacun des deux items (45 valeurs),
- une décision de degré de similitude donnée par un tribunal ou un office d’enregistrement de marques,
- le tribunal ou l’office d’enregistrement des marques ayant rendu le verdict (17 valeurs),
- l’origine de la décision, e.g. Tribunal de justice, Chambre de recours, Pratique de l’office,... (6 valeurs).

Une information supplémentaire est disponible dans le jeu de données concernant le raisonnement justifiant la décision. L’objectif étant d’automatiser le travail de l’expert, cette information n’a pas été retenue pour la suite de l’étude mais sera utilisée pour expliquer certains résultats.

Le degré de similitude donnée par l’office des marques comprend 6 niveaux : Identique, Degré élevé, Similaire, Degré faible, Dissemblable. L’objectif est d’aider l’expert en proposant un premier tri automatique permettant de détecter une possible similitude, et ce, quel que soit son degré. Pour cela nous créons une variable Similitude avec deux

| Nom du modèle | Performance | Taille (Mo) |
|----------------------|-------------|-------------|
| all-mpnet-base-v2 | 69.57 | 420 |
| all-distilroberta-v1 | 68.73 | 290 |
| all-MiniLM-L12-v2 | 68.70 | 120 |
| all-MiniLM-L6-v2 | 68.06 | 80 |

TABLE 1 – Modèles d’*embeddings*

classes :

- Semblable si la décision est Identique, Degré élevé, Similaire, Degré faible
- Dissemblable sinon

Les deux classes sont déséquilibrées : Semblable = 15846, Dissemblable = 91724. Pour équilibrer les classes, 15846 décisions sont choisies aléatoirement dans la classe Dissemblable. Le jeu de données est ainsi réduit à 31692 décisions avec autant d’observations dans chacune des deux classes. L’objectif est de construire un modèle permettant de déterminer la variable Similitude à partir des autres caractéristiques du jeu de données.

3 Similarité sémantique

Les items décrivant les services ou les produits sont des données textuelles qu’il est nécessaire de traiter au préalable. Une approche classique consiste à utiliser une représentation vectorielle des items (*embeddings*) puis à en extraire une mesure de similarité. La plus utilisée est la mesure de similarité cosinus,

$$\text{cosinusSim}(E_{t_i}, E_{t_j}) = \frac{E_{t_i} \cdot E_{t_j}}{\|E_{t_i}\| \|E_{t_j}\|},$$

où E_{t_i} et E_{t_j} sont les *embeddings* des paires d’items t_i et t_j présentes dans le jeu de données.

Concernant l’*embedding* des items, il existe une multitude de modèles. Nous avons mené une étude comparative de différents modèles afin de déterminer leur impact sur la mesure de similarité des items. Nous avons sélectionné quatre modèles (distilroberta, llm_minilm6, minilm12, mpnet) depuis la librairie *SentenceTransformers* issue des travaux de [6]. Ils ont été pré-entraînés sur plus d’un milliard de paires d’apprentissage. Ils sont considérés comme les plus performants pour la tâche de vectorisation d’une phrase et ils sont de taille variable comme le montre le tableau 1.

Nous avons également testé deux autres modèles plus légers, word2vec (w2v) pré-entraîné sur GoogleNews [4] et un autre LLM, minilm3, entraîné à détecter des paraphrases sur un jeu de données de moins de 100 millions de paires, et ne faisant que 61 Mo. Le modèle word2vec donnant des *embeddings* de mots, nous avons pris la moyenne des vecteurs des mots constituant l’item.

La figure 1 montre que tous les modèles permettent d’obtenir une mesure de similarité cosinus plus élevée pour la classe Semblable, ce qui confirme qu’il est pertinent d’utiliser cette mesure pour discriminer les items semblables et dissemblables. L’étalement du cosinus pour la classe Semblable est certainement dû au regroupement des classes initiales (Identique, Similaire, Degré élevé, Degré faible).

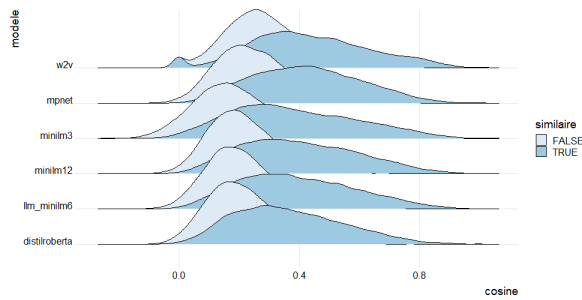


FIGURE 1 – Distribution du cosinus en fonction du modèle et des classes Semblable (TRUE) et Dissemblable (FALSE)

Pour choisir le meilleur *embedding*, on compare leur score de prédiction en comparant la classe observée et la classe obtenue en seuillant le cosinus. Les figures 2, 3, 4 et 5 montrent les scores obtenus en fonction du seuil.

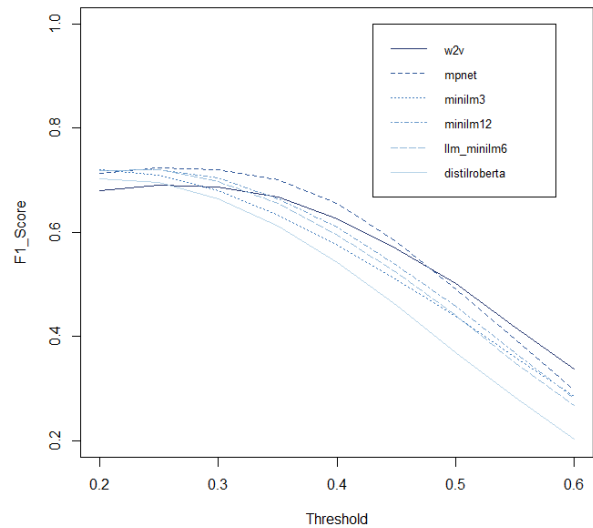


FIGURE 3 – F1_score en fonction du seuil

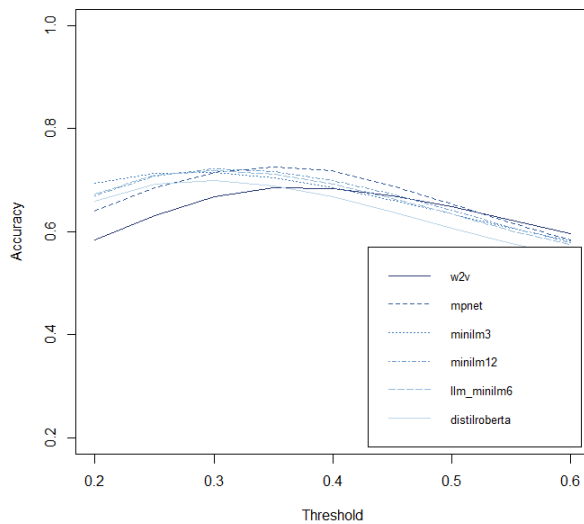


FIGURE 2 – Exactitude en fonction du seuil

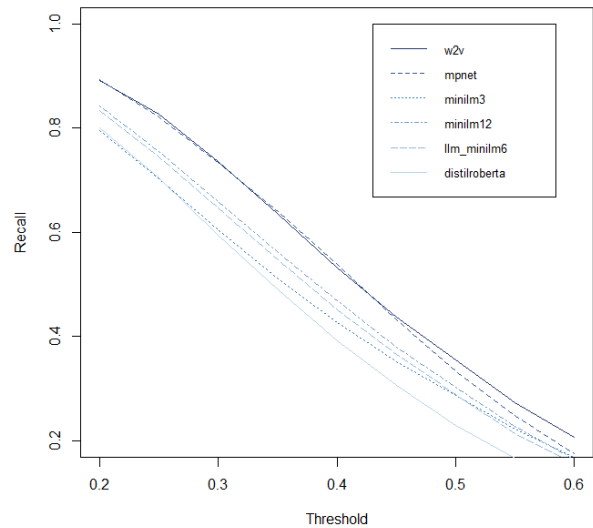


FIGURE 4 – Rappel en fonction du seuil

On note que les modèles w2v et mpnet donnent un taux de bien classés légèrement meilleur pour la classe Semblable (*Recall*) mais sont nettement moins performants pour la classe Dissemblable (*Specificity*). Or seuls les items détectés semblables bénéficieront d'un traitement manuel. Il est donc primordial de bien prédire les items dissemblables car ils seront écartés de l'étude. Les autres modèles *Transformers* donnent des résultats identiques. On retiendra le plus léger et rapide minilm3, qui nous permettra de comparer efficacement un grand volume de paires d'items.

Le meilleur compromis entre taux de bien classés pour la classe Semblable et la classe Dissemblable est atteint pour un seuil de 0.275 comme le montre la figure 6.

Avec ce seuil, le modèle minilm3 permet d'atteindre un score de 70% de bien classés quelle que soit la classe (Ligne "Sans modèle" du tableau 2).

4 Modèles d'apprentissage

La détection d'items similaires repose essentiellement sur la similitude sémantique des items mais peut aussi être influencée par des paramètres complémentaires tels que l'office qui a rendu la décision ou les catégories de la classification de Nice des items. L'objectif de ce paragraphe est de déterminer un modèle d'apprentissage de classification de la variable Similitude avec en entrée la mesure de similarité cosinus, l'office, l'origine, les catégories de la classification de Nice des items 1 et 2 et une nouvelle variable binaire indiquant si la catégorie est identique pour les deux items. Le schéma du processus mis en place est donné dans la figure 7.

Les variables office, origine, catégorie de la classification de Nice de l'item 1 et de la classification de Nice de l'item 2 sont catégorielles avec 6, 17, 45 et 45 modalités respec-

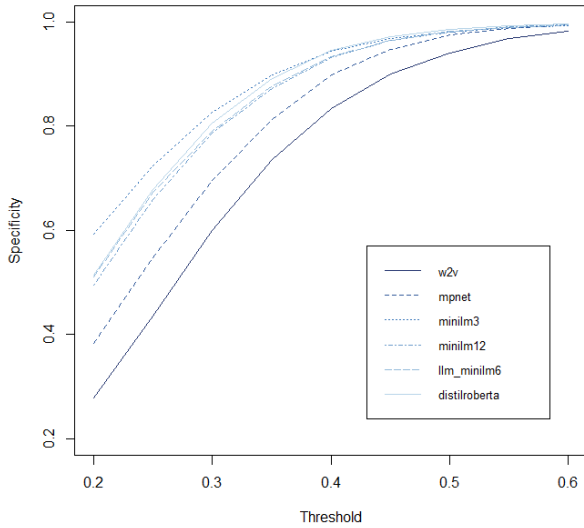


FIGURE 5 – Specificité en fonction du seuil

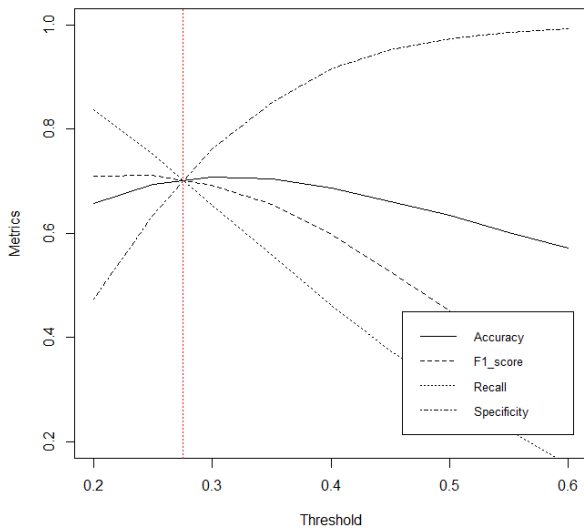


FIGURE 6 – Métriques en fonction du seuil pour le modèle minilm3

tivement. Chaque modalité est encodée en binaire pour obtenir un nombre total de 115 variables en entrée. Il existe un descriptif textuel des catégories de la classification de Nice. Afin de ne pas alourdir le modèle, nous avons préféré ne pas utiliser de modèle d'*embedding* et de traiter ces variables comme catégorielles.

Les modèles d'apprentissage utilisés sont un arbre de décision avec une profondeur maximale de 10, une régression logistique, une forêt aléatoire avec 100 arbres et 2 variables testées à chaque nœud, un réseau de neurones (MLP) avec une couche cachée constitué de 5 neurones (586 poids) et un autre avec 15 neurones (1756 poids) avec la fonction d'activation sigmoïde.

Les modèles sont construits sur une base d'apprentissage contenant 70% du jeu de données choisi aléatoirement, et sont comparés sur une base de test avec les 30% restants.

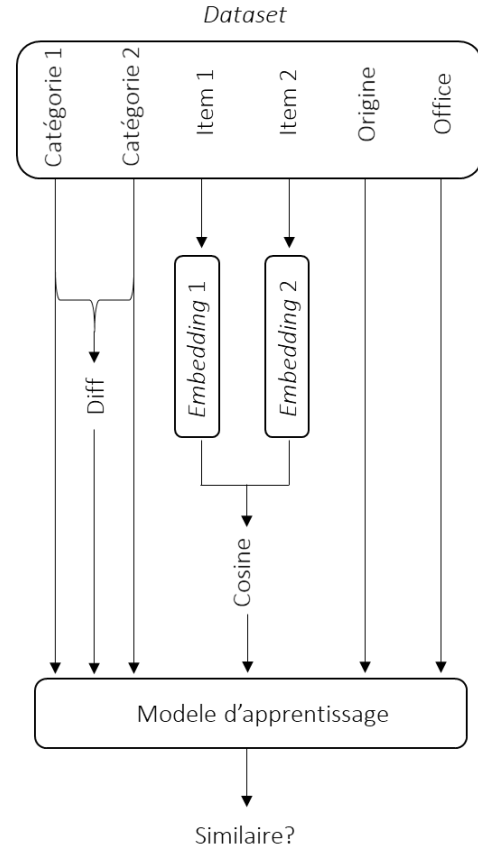


FIGURE 7 – Schéma général

Il y a trois niveaux d'aléa dans le processus : au niveau du re-équilibrage des classes, lors de la construction des bases d'apprentissage et de test, le caractère stochastique de certains modèles (Forêts aléatoires, MLP). Afin de quantifier l'incertitude due à ces aléas et vérifier la robustesse des résultats, le processus est répété 20 fois. Les figures 8, 9, 10 et 11 représentent la distribution des 20 métriques obtenues pour chaque modèle et le tableau 2 donne leurs valeurs moyennes. Les boxplots permettent d'appréhender la robustesse des résultats qui restent très stables à l'exception du rappel et de la spécificité par l'arbre de décision.

| Modèle | Exact | F1 | Rappel | Spec |
|-----------------------|-------|------|--------|------|
| Sans modèle | 0.71 | 0.71 | 0.71 | 0.70 |
| Arbre de décision | 0.79 | 0.78 | 0.76 | 0.81 |
| Régression logistique | 0.83 | 0.83 | 0.81 | 0.85 |
| Forêt aléatoire | 0.85 | 0.85 | 0.84 | 0.86 |
| MLP (5) | 0.84 | 0.84 | 0.83 | 0.85 |
| MLP (15) | 0.84 | 0.84 | 0.83 | 0.85 |

TABLE 2 – Tableau des valeurs moyennes des métriques

A l'exception de l'arbre de décision, tous les modèles permettent d'améliorer le résultat obtenu sans modèle de 10 à 15% sur toutes les métriques. Le test statistique de l'analyse de la variance montre clairement qu'il n'y a pas de différence significative pour les différentes métriques entre

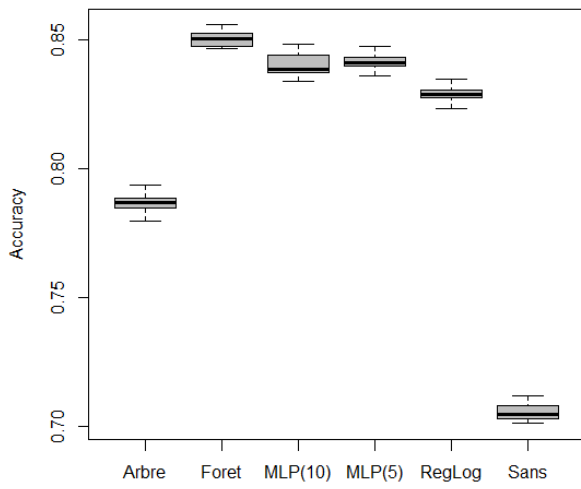


FIGURE 8 – Boxplots de l'exactitude

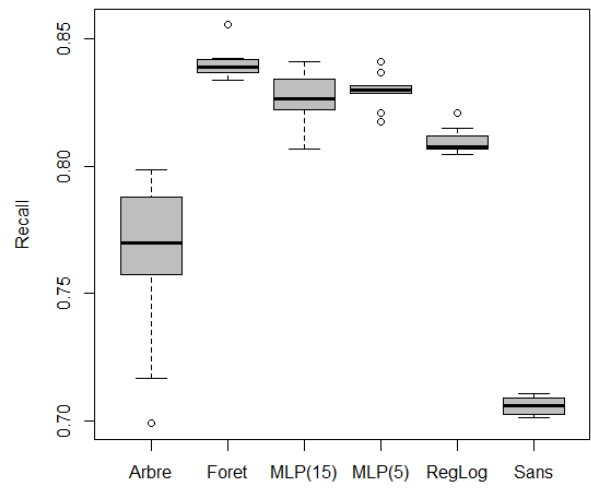


FIGURE 10 – Boxplots du rappel

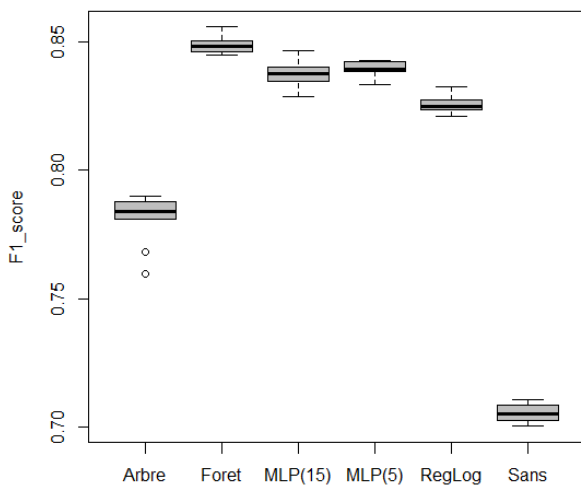


FIGURE 9 – Boxplots du F1_score

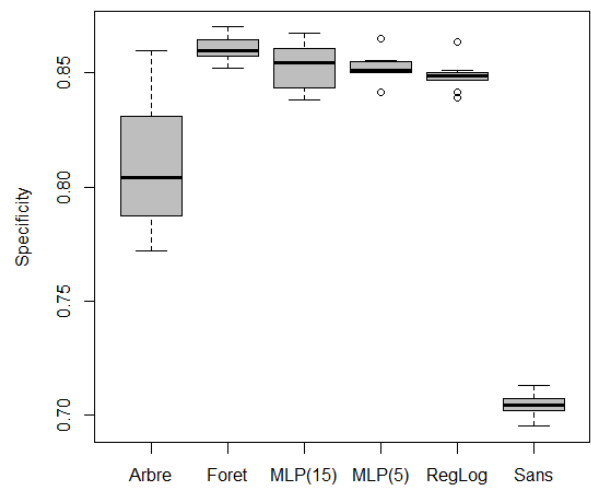


FIGURE 11 – Boxplots de la Specificité

la régression logistique, la forêt aléatoire et les deux réseaux de neurones (p-valeur inférieure à 10^{-11}). Cependant graphiquement, on peut voir des métriques légèrement plus élevées pour la forêt aléatoire. De plus ce modèle a l'avantage d'être partiellement interprétable (figure 12). Bien que moins performants que la régression logistique, la forêt aléatoire ou les réseaux de neurones, l'arbre de décision améliore aussi les métriques. C'est un modèle qui reste intéressant car il permet d'expliquer entièrement la classification d'une nouvelle instance.

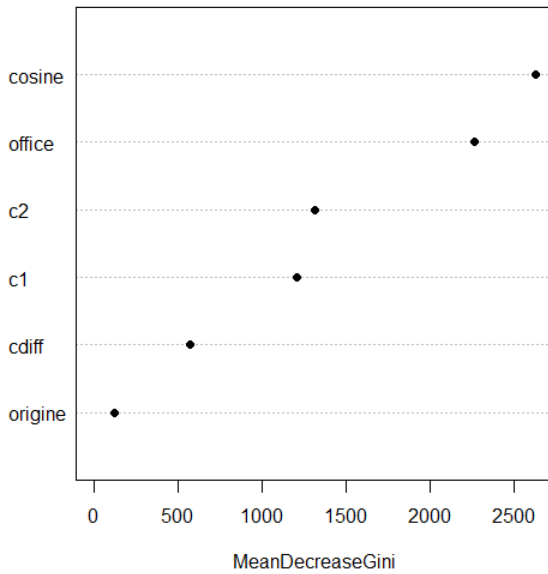


FIGURE 12 – Importance des variables dans la forêt aléatoire

5 Analyse des résultats

Impact de l'office. L'importance des variables de la forêt aléatoire montre que les variables déterminantes pour la classification sont le cosinus et l'office, suivies dans une moindre mesure par les catégories de la classification de Nice des deux items (figure 12). Le fait que la variable office ait une importance élevée pourrait indiquer un biais selon qui a rendu le jugement. Nous avons donc procédé à une validation croisée en entraînant le modèle sur 16 offices et en testant sur le 17ème office. Les résultats présentés dans la figure 13 montrent que cela n'est pas le cas. Ils sont à peu près stables sauf pour la Hongrie, l'Estonie et la Bulgarie. Ces derniers présentent une dissymétrie entre le rappel et la spécificité avec un rappel assez bas (50%-70%). Cela peut s'expliquer par une arrivée plus tardive de ces pays dans l'Union Européenne, donc moins de temps pour harmoniser les pratiques.

Erreur selon la classification de Nice. La répartition des erreurs au sein des catégories de la classification de Nice (Figure 14) montre un déséquilibre.

Cinq catégories présentent une erreur inférieure à 5% dont les trois meilleures : 15 (*Musical instruments; music stands and stands for musical instruments; measuring sticks*), 27

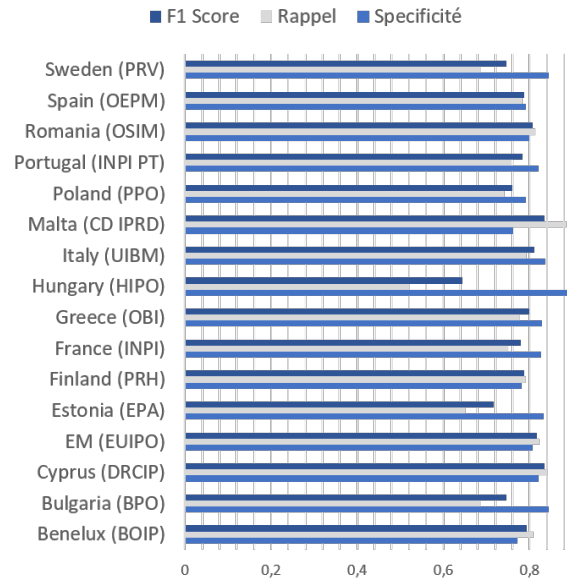


FIGURE 13 – Métriques par Office

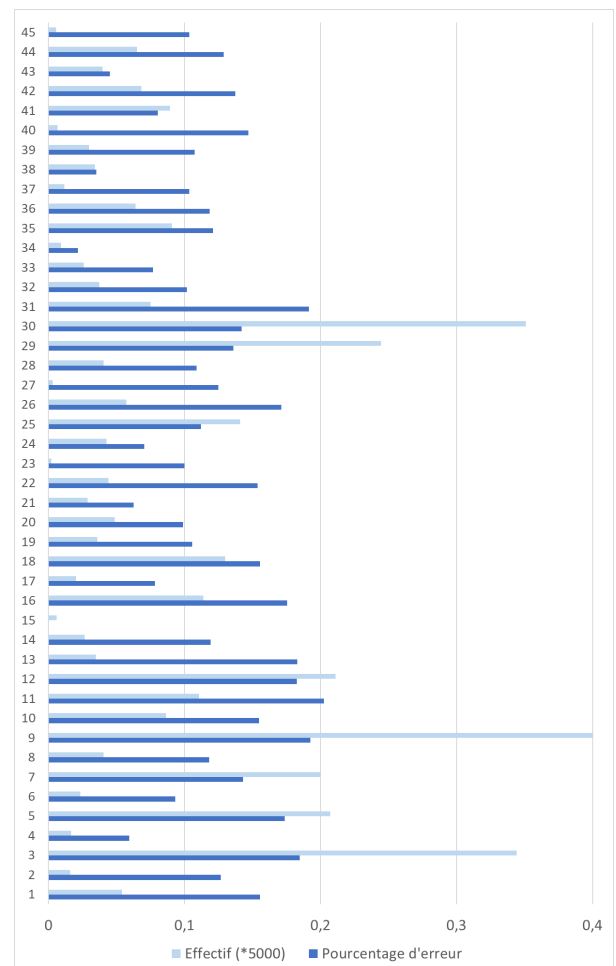


FIGURE 14 – Pourcentage d'erreur par catégorie

(*Carpets, mats, linoleum and other floor coverings; wall hangings, not of textile materials*), 34 (*Tobacco and to-*

bacco substitutes; cigarettes and cigars; electronic cigarettes and smokers' sprays; smokers' articles; matches). En revanche, une dizaine de catégories ont un taux d'erreur supérieur à 15%, dont les trois premières : 9 (*Scientific, research, navigational, geodetic, photographic, cinematographic, audiovisual, optical, weighing, measuring, signaling, detection, testing, inspection, rescue and teaching apparatus and instruments; apparatus and instruments for conducting, distributing, transforming, accumulating, regulating or controlling the distribution or consumption of electricity; apparatus and instruments for recording, transmitting, reproducing or processing sound, images or data; recorded or downloadable media, software, blank digital or similar recording and storage media; mechanisms for prepayment machines; cash registers, calculating devices; computers and computer peripherals; diving suits, diving masks, diving ear plugs, nose clips for divers and swimmers, diving gloves, breathing apparatus for underwater swimming; fire extinguishers.*), 11 (*Appliances and installations for lighting, heating, cooling, steam production, cooking, drying, ventilation, water distribution and sanitary installations*), 31 (*Raw and unprocessed agricultural, aquacultural, horticultural and forestry products; raw and unprocessed seeds; fresh fruit and vegetables, fresh herbs; natural plants and flowers; bulbs, seedlings and seeds; live animals; animal feed and beverages; malt.*). Les catégories présentant un fort taux d'erreur semblent recouvrir un plus large champ de produits ou de services que celles avec une faible erreur. Cela pourrait expliquer les difficultés du modèle à être performant sur des catégories très larges. Une solution pourrait être d'entraîner un modèle spécifique à chacune de ces catégories.

Erreur selon le degré de similitude. Pour les items de la classe Semblable, le taux d'erreur du modèle de forêt aléatoire est 16% (table 2). Si on examine plus en détail ce taux selon la classe d'origine (avant regroupement), on observe qu'il est naturellement proportionnel au degré de similitude (Figure 15).

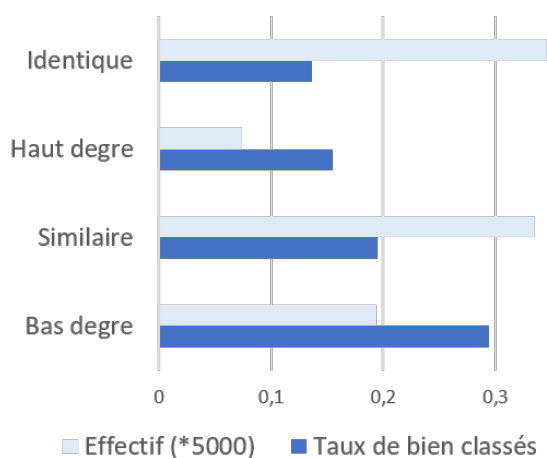


FIGURE 15 – Pourcentage d'erreur par degré de similitude
Dans la classe Identique le taux d'erreur n'est pas nul (6%)

car il ne s'agit pas d'items identiques syntaxiquement mais désignant un même produit ou service comme par exemple, *Marketing campaigns* et *Television advertising*, ou *Articles for smokers* et *Cigarette filters*. Le taux d'erreur atteint 29% pour la classe Bas degré. Il pourrait être intéressant d'utiliser un modèle d'apprentissage multiclasse et ne pas procéder au regroupement initial des classes. Il n'est pas certain qu'un tel modèle améliore le taux d'erreur de la classe Bas degré car celui-ci traduit une incertitude du décideur pouvant dépendre du contexte. Par exemple *Travelling bags* et *Bags for sports* ont été jugés de faible degré de similitude car « les sacs de voyage sont essentiellement des valises et ne couvrent donc pas les sacs de sport » mais qu'ils ont tout de même des points de similitude : « 1. Nature, 3. Méthodes d'utilisation, 6. Canaux de distribution, 8. Producteur / Fournisseur ».

6 Conclusion

Dans ce travail, nous avons proposé un outil d'aide à la décision pour l'opposition de marques avec une détection automatique de la similitude entre items. Cette approche présente deux originalités. L'utilisation de *sentence embeddings* basés sur les *Transformers* a amélioré la détection de similarité sémantique des items habituellement calculée à partir de *word embeddings*. L'intégration de la mesure de similarité cosinus avec d'autres caractéristiques sur les décisions dans un modèle d'apprentissage a augmenté de 10 à 15% les différentes métriques de performance de classification. Il demeure cependant un certain nombre d'erreurs qui ne peuvent s'expliquer ni par une proximité sémantique, ni par une autre caractéristique. Par exemple, les items *Preserved fruit* et *Rice-based snacks* sont détectés dissemblables avec notre modèle mais ils sont considérés comme similaires par l'office d'enregistrement des marques. La décision est argumentée par le fait que les deux peuvent être essentiellement des en-cas. Afin de palier ce problème, nous envisageons dans de futurs travaux de *fine-tuner* un LLM avec notre jeu de données, et de tester des *embeddings* plus frugaux, que nous pourrions créer à partir de notre jeu de données dans une approche similaire à SINr [5].

Références

- [1] Hayfa Alshowaish, Yousef Al-Ohali, and Abeer Al-Nafjan. Trademark image similarity detection using convolutional neural network. *Applied Sciences*, 12(3), 2022.
- [2] Jiangzhong Cao, Yunfei Huang, Qingyun Dai, and Wing-Kuen Ling. Unsupervised trademark retrieval method based on attention mechanism. *Sensors*, 21(5), 2021.
- [3] Yingchi Liu, Quanzhi Li, Changlong Sun, and Luo Si. Similar trademark detection via semantic, phonetic and visual similarity information. In *SIGIR '21: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2025–2030, July 2021.

- [4] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.
- [5] Thibault Prouteau, Victor Connes, Nicolas Dugué, Anthony Perez, Jean-Charles Lamirel, Nathalie Camelin, and Sylvain Meignier. SINr : Fast Computing of Sparse Interpretable Node Representations is not a Sin! In *Advances in Intelligent Data Analysis XIX, 19th International Symposium on Intelligent Data Analysis, IDA 2021*, number 12695 in Lecture Notes in Computer Science, pages 325–337, Porto, Portugal, April 2021. Springer, Cham. Frontier Prize (best paper).
- [6] Nils Reimers and Iryna Gurevych. Sentence-bert : Sentence embeddings using siamese bert-networks. In *Conference on Empirical Methods in Natural Language Processing*, 2019.
- [7] Daniyil Shmatkov, Oleksii Gorokhovatskyi, and Nataliya Vnukova. Elaborative trademark similarity evaluation using goods and services automated comparison. In *Computational Linguistics and Intelligent Systems 2023 (Volume III)*, Kharkiv, Ukraine, April 2023.
- [8] Amy J.C. Trappey, Charles Trappey, and Sam C.C. Lin. Detecting trademark image infringement using convolutional neural networks. In *Advances in Transdisciplinary Engineering*, October 2019.
- [9] Amy J.C. Trappey, Charles V. Trappey, and Samuel Shih. An intelligent content-based image retrieval methodology using transfer learning for digital ip protection. *Advanced Engineering Informatics*, 48 :101291, 2021.
- [10] Charles V. Trappey, Amy J.C. Trappey, and Sam C.-C. Lin. Intelligent trademark similarity analysis of image, spelling, and phonetic features using machine learning methodologies. *Advanced Engineering Informatics*, 45 :101120, 2020.
- [11] Osman Tursun, Simon Denman, Sabesan Sivapalan, Sridha Sridharan, Clinton Fookes, and Sandra Mau. Component-based attention for large-scale trademark retrieval. *IEEE Transactions on Information Forensics and Security*, 17 :2350–2363, 2022.
- [12] Julie Tytgat, Guillaume Wisniewski, and Adrien Bétrancourt. Apprentissage automatique pour la surveillance de marques. In *Conférence Nationale sur les Applications Pratiques de l’Intelligence Artificielle*, Saint-Etienne, France, June 2021.
- [13] Dmitry Vesnin, Dmitry Levshun, and Andrey Chechulin. Trademark similarity evaluation using a combination of vit and local features. *Information*, 14(7), 2023.