Classification de phrases courtes : des approches non-supervisées aux approches faiblement supervisées

Kaoutar Ghazi, Sébastien Marchal, Andon Tchechmedjiev, Pierre-Antoine Jean, Nicolas Sutton-Charani, Sébastien Harispe

Classification de phrases courtes : des approches non-supervisées aux approches faiblement supervisées

Illustration sur une problématique industrielle

Kaoutar Ghazi*, Sébastien Marchal*, Andon Tchechmedjiev*
Pierre-Antoine Jean*, Nicolas Sutton-Charani*, Sébastien Harispe*

*LGI2P, IMT Mines Alès, Univ. Montpellier, Alès, France
prénom.nom@mines-ales.fr

Résumé. Cette note présente une étude, menée dans un contexte industriel, de différentes approches de classification non supervisée ou faiblement supervisée de courtes requêtes exprimées en langage naturel. Nous présentons et comparons différentes approches basées à la fois sur des techniques de Recherche d'Information et sur des techniques à base d'apprentissage machine exploitant différents types de plongements sémantiques (*embeddings*). Nous discutons les résultats obtenus avant d'élargir par une présentation d'approches alternatives à base d'apprentissage automatique supervisé ne nécessitant que peu de données labelisées – des approches de type *few shot learning*. Cette note vise ainsi à faire synthèse des approches dites état de l'art qui peuvent être utilisées pour traiter cette problématique de classification fréquemment rencontrée dans l'Industrie (e.g., chatbot).

1 Introduction

De nombreuses problématiques de Traitement Automatique des Langues qui sont d'intérêt pour l'Industrie peuvent être formulées ou approchées comme des problématiques de classification de textes courts, e.g. ChatBots, analyse de tweets ou d'avis client lors d'analyses d'opinions. Cette note se concentre sur le contexte que nous avons rencontré en pratique, potentiellement fréquent dans l'industrie, dans lequel peu ou pas de données de domaine labélisées sont disponibles lors de la définition des modèles de résolution de la tâche de classification, i.e. seules les classes d'intérêt décrites par un texte court, et éventuellement quelques exemples labelisées sont fournis.

Des modèles classiques de Recherche d'Information, aux modèles à base d'apprentissage machine, nous présentons dans cette note une synthèse de différentes approches éprouvées de l'état de l'art et outillées, qui peuvent aujourd'hui être facilement mises en oeuvre pour aborder ce type de problématiques de classification, notamment dans l'Industrie. Nous illustrons plus particulièrement l'utilisation de ces approches dans un contexte correspondant à la classification de requêtes exprimées sous la forme de textes courts (une à deux phrases), dans lequel nous disposons *a priori* seulement d'un court descriptif en langage naturel pour chaque classe.

Nous distinguerons par la suite deux cas : (i) *cold-start* – pas de donnée labélisée, (ii) *few-shot* – peu de données labélisées.

Il est en effet fréquent dans l'Industrie d'être confronté au cas où : (a) seules les classes et leurs descriptions sont fournies dans un premier temps avec peu ou pas de données labélisées, (b) une labélisation des données est éventuellement effectuée suite au déploiement en production du modèle initial, e.g. en analysant les données traitées par le système déployé. On se retrouve alors ensuite dans un contexte d'apprentissage dit en ligne (*online learning*). Cette note se concentre donc sur la phase (a) précitée et est structurée comme suit.

Nous présentons dans un premier temps les approches étudiées pour élaborer un système de classification dans les contextes non-supervisés (problématique *cold start*) et faiblement supervisés. L'objectif n'est pas ici d'être exhaustif, mais plutôt de discuter des approches aujourd'hui simples à mettre en oeuvre en pratique – a minima dans un contexte R&D. Nous présentons et discutons par la suite les résultats obtenus sur un jeu de données spécifique. Des perspectives pour la mise en place de systèmes plus raffinés sont par la suite proposées.

2 Approches étudiées

La problématique que nous traitons dans cette note correspond à la classification de textes courts (requêtes), étant donné un descriptif textuel de chacune des classes. Nous présentons ci-après les approches considérées pour traiter cette problématique.

2.1 Approches Recherche d'Information (RI)

Les approches que nous avons étudiées pour la classification de requêtes, sans données labélisées, se basent sur des méthodes éprouvées en Recherche d'Information notamment celles qui reposent sur l'utilisation de stratégies de pondération qui visent à assigner un poids aux termes afin d'indiquer leur importance thématique dans un document.

L'importance d'un terme est généralement mesurée par rapport à la fréquence de ses occurrences ou co-occurences dans un contexte donné (fenêtre glissante, une phrase, un paragraphe, un document), qu'il s'agisse du nombre de ses occurrences dans un document, de son TF-IDF [1] par rapport à un document, du nombre de ses co-ocurrences dans l'ensemble des documents, ou de son PPMI (*Positive Pointwise Mutual Information*) [2] avec un autre terme dans le corpus.

Matrice terme-document (TF-IDF)

Analyser l'importance des termes décrivant une classe peut être utile pour notre problème de classification d'une requête. Comme nous procédons par une classification non-supervisée, nous proposons de calculer la matrice terme-document pour chaque requête telle que l'ensemble des documents correspondra aux descriptifs des classes plus la requête. Les éléments de la matrice sont le TF-IDF d'un mot dans un document. Nous obtenons par la suite une représentation des classes et de la requête (vecteurs colonnes de la matrice). Enfin, nous déduisons

1. $tf-idf(m,d) = tf(m,d) * idf(m)$, avec $tf(m,d)$ la fréquence du mot m dans le document d et $idf(m) = \log(\frac{n}{df(m)}) + 1$ t.q. n le nombre total de document, $df(m)$ ceux contenant m.

2. $PPMI(m_1, m_2) = max(log_2(\frac{P(m_1,m_2)}{P(m_1)P(m_2)}, 0)$ avec $P(m_1, m_2)$ est la probabilité d'observer les mots m_1 et m_2 ensemble et $P(m_1)$ la probabilité d'observer m_1 indépendemment de m_2.

la classe d'appartenance de la requête en mesurant la similarité (euclidienne, cosinus ou corrélation) entre sa représentation et celle de chacune des classes puis en gardant la classe la plus proche en terme de distance. Pour illustrer cette approche, nous supposons avoir m classes, $C_1 \dots C_m$, et une requête R qu'on souhaite classifier. Nous calculons alors la matrice terme-document, sur le vocabulaire constitué des mots des descriptifs des m classes et de la requête R (voir le tableau ci-dessous), afin d'avoir une représentation des classes et de la requête par un vecteur de nombres réels $[tf\_idf_{1,i}, \dots, tf\_idf_{n,i}]$ pour i allant de 1 à m ou $i = R$. La classe C_k de R est celle dont la distance entre sa représentation et celle de la requête vaut

$$min\{d([tf\_idf_{1,i}, \dots, tf\_idf_{n,i}], [tf\_idf_{1,R}, \dots, tf\_idf_{n,R}]) | 1 \leq i \leq m\}.$$

Vocabulaire	C_1	C_2	C_3	...	C_m	R
m_1	tf_idf$_{1,1}$	...	...	...	tf_idf$_{1,m}$	tf_idf$_{1,R}$
m_2	...	...	...	...	...	...
m_3	...	...	...	...	...	...
m_4	...	...	...	...	...	...
m_5	...	...	...	...	...	...
...	...	...	...	...	...	...
m_n	tf_idf$_{n,1}$	...	...	...	tf_idf$_{n,m}$	tf_idf$_{n,R}$

Matrice terme-terme (PPMI)

Suivant le même procédé, nous proposons de calculer la matrice terme-terme pour chaque requête telle que les éléments de la matrice correspondent aux PPMI de deux mots dans le corpus constitué des descriptifs des classes plus la requête. Nous calculons par la suite une représentation des classes et de la requête par une agrégation (souvent une moyenne, mais il existe des techniques état de l'art de reprojection [3]) des vecteurs représentant leur mots (vecteurs lignes de la matrice). Enfin, la classe d'appartenance de la requête est celle la plus proche en terme de distance entre sa représentation vectorielle et la représentation de la requête en question.

2.2 Approches par plongement de mots

Au-delà des approches distributionnelles mentionnées ci-avant (e.g. plongement de mots à base de PPMI), nous avons opté pour des représentations sémantiques plus génériques des mots issues de plongements pré-entraînés (plongements classiques, ou modèles de langue contextualisés) : les plongements CBOW (resp. SkipGram) qui sont appris par l'architecture neuronale de word2vec et qui permettent de prédire un mot sachant son contexte (resp. prédire le contexte étant donné un mot), puis les plongements FastText qui suivent la même procédure que SkipGram, mais avec des n-grammes au lieu des tokens. Un n-gramme est une séquence de n caractères consécutifs d'un mot, e.g. un mot à 4 lettres est composé de trois 2-grammes.

Dans cette note, nous considérons des plongements pré-calculés disponibles en ligne [4] pour la langue Française : 1/ plongements CBOW appris sur le corpus WaC et de dimension 500 (taille du vecteur de plongement), 2/ plongements SkipGram appris sur le corpus WaC et de

3. https ://github.com/zalandoresearch/flair
4. https ://fauconnier.github.io/data

dimension 500, 3/ plongements CBOW appris sur le corpus Wikipédia et de dimension 700, 4/ plongements SkipGram appris sur le corpus Wikipédia et de dimension 1000.

De plus, nous générons des plongements FastText à l'aide de la libraire Flair Akbik et al. (2019), qui sont appris sur Wikipédia et de dimension 300.

Cette approche basée plongement de mots, nous permettra de faire une classification par rapprochement sémantique d'une requête aux descriptifs des classes.

2.3 Approches à base d'apprentissage supervisé

En nous basant sur la librairie Flair, nous avons également mis en place une architecture de classification de textes supervisée, avec les plongements contextualisés les plus récents pour la langue Française – CamemBert Martin et al. (2019) – en entrée, qui alimentent un encodeur (vecteurs de documents) à base d'un réseau profond récurrent (biGRU), qui lui même alimente une couche de classification linéaire. Nous explorons les deux stratégies : cold start, en prenant les descriptifs des classes sur-échantillonées (209 $\times$ nombre de classes) comme un jeu d'entraînement ; *few-shot learning*, en considérant, dans un premier temps, 90% des requêtes que nous disposons pour les évaluations comme jeu d'entraînement (les 10% restant serviront pour les tests), puis dans un second temps, en combinant les descriptifs des classes avec des données labélisées (90% des requêtes extraites du jeu de données de test) pour constituer la base d'apprentissage. Dans ces derniers cas, nous évaluons les performances du système avec une validation croisée (10 divisions) que nous détaillons dans la suite.

3 Résultats

3.1 Description des données

Nous avons testé les approches décrites auparavant sur un jeu de données issue de la mairie de Chatou. Les classes représentent les motifs de visite de la mairie (e.g. Conseils budgétaires) et sont à l'ordre de 18 classes. La plupart des classes possèdent un descriptif genre : Le crédit municipal de Paris vous aide à analyser vos dépenses et recettes, vous renseigne sur les différentes aides existantes. Les requêtes sont exprimées en language quotidien e.g. « Je débute dans l'entreprenariat j'aurais besoin dans suivi budgétaire »et sont à l'ordre de 180 requêtes (10 requêtes par classe).

3.2 Évaluations

Nous évaluons la performance des approches considérées dans cette note par la précision qui correspond au nombre de requêtes que le système a réussi à classifier sur le nombre total des requêtes (taille du jeu de données de test).

Pour l'approche basée sur l'apprentissage supervisé par un échantillon du jeu de données de tests (*few-shot learning*), nous avons opté pour une validation croisée qui consiste à diviser les données dont ont dispose pour faire les évaluations en deux sous-échantillons aléatoires ; le premier pour l'apprentissage (90% comme évoqué auparavant) et le second pour le test (10%) puis mesurer la performance du système pour cette division et enfin répéter l'opération 10 fois. Nous mesurons la performance de cette approche par la moyenne des 10 précisions calculées.

3.3 Analyse

Les résultats de classification obtenus pour les différentes approches sur notre jeu de données montrent que le mode *cold start* demeure une solution non négligeable pour une classification de textes courts, étant donné les descriptifs des classes, si aucune donnée de supervision n'est disponible. Le meilleur score est obtenu grâce aux plongements CBOW entraînés sur le corpus Wikipédia en utilisant la correlation (ou même le cosinus) comme une mesure de similarité.

En effet, les plongements FastText étant d'une dimension réduite (300 vs $dim \geq 500$ pour CBOW et SkipGram), cela peut justifier les mauvais résultats obtenus en considérant ces plongements par rapport aux autres appris par CBOW et SkipGram. Même si les deux techniques CBOW et SkipGram sont quasiment similaires, il s'avèrent que CBOW est plus intéressante pour notre jeux de données (Voir CBOW$_{WaC}$ vs SkipGram$_{WaC}$). En comparant les résultats de CBOW$_{WaC}$ et CBOW$_{Wiki}$ (resp. SkipGram$_{WaC}$ et SkipGram$_{Wiki}$), nous observons l'impact de la nature du corpus d'apprentissage des plongements des mots sur les résultats obtenus. Pour les plongements CBOW, le corpus Wikipédia est plus adéquat pour capturer le sens des termes de notre vocabulaire métier que le corpus WaC dont le contenu correspond à des articles tirés du Monde Diplomatique. Nous observons que la technique TF-IDF est aussi compétitive dans notre contexte.

Pour les représentations calculées avec $PPMI$, malgré qu'elles capturent le sens des termes dans leur contexte métier (corpus = Descriptifs des classes + requête) qui est plus spécifique par rapport à leur représentation Word2Vec qui sont appris sur des contextes plus génériques, nous concluons que ce raisonnement n'est pas approprié. Nous pourrons justifier ceci par le fait que les requêtes ont des représentations $PPMI$ par des vecteurs creux (les termes d'une requête co-occurent rarement avec les termes des descriptifs des classes) ce qui n'est pas le cas pour les vecteurs représentants les classes, par conséquent ceci pourrait induire à des erreurs de classification. Notant qu'avec des représentations TF-IDF, nous obtenons des représentations par des vecteurs creux pour les requêtes mais aussi pour les classes grâce à la pénalisation des termes fréquents.

Distance	TF-IDF	PPMI	FastText	CBOW		SkipGram	
				WaC	Wiki	WaC	Wiki
Euclidienne	44%	5%	31%	32%	26%	27%	27%
Cosinus	44%	5%	27%	44%	55%	39%	37%
Correlation	46%	5%	31%	44%	**56**%	40%	37%

TAB. 1 – *Les résultats des différentes approches sans données de supervision*

Nous observons qu'une approche basée sur une classification supervisée par les descriptifs des classes donne des résultats de même grandeur que ceux obtenus suivant une approche non-supervisée. Cependant, avec peu de données annotées, nous avons pu doubler les performances du classifieur. Rappelant que ces évaluations corespondent à une validation croisée (10-fold).

Finalement, le classifieur Flair semble ne pas être aussi pertinent que les meilleures approches précitées, mais en procédant en mode *few-shot learning*, ou suivant une stratégie d'augmentation des données par les descriptifs des classes, nous arrivons à améliorer considérablement les résultats de la classification. Notons que nous avons appliqué une phase de

Corpus	Précision
Descriptifs des classes	39%
180 requêtes	95%
Descriptifs des classes + 162 requêtes	95%

TAB. 2 – *Les résultats de l'approche faiblement supervisée*

pré-traitement préalable (lemmatisation et suppression des mots vides et des ponctuations) sur nos données textuelles. La suppression des mots vides n'est pas toujours une bonne solution pour notre problème de classification surtout que ceci peut induire à des erreurs de classification, comme par exemple la demande suivante « je viens récupérer le document ~~que~~ j'ai déposé »correspond à un retrait mais pas à un dépôt. Cependant, les problématiques pouvant être associées à la suppression des mots vides n'étant pas mis en évidence sur notre jeu de données, nous justifions ainsi que les meilleurs résultats que nous avons obtenus coïncident avec ceux sur lesquels nous avons réalisé une phase de pré-traitement complète.

4 Élargissement

Les résultats de classification dans notre contexte ne dépassent pas 60% avec les approches non supervisées basées sur l'utilisation de techniques de recherche d'information ou de plongement de mots. Les résultats que nous avons obtenus en testant une autre méthode de pondération, le « BM25 »développé par Robertson et Sparck Jones, sont aussi de même ordre de grandeur que le TF-IDF. Nous pensons qu'en combinant la méthode de pondération avec les plongements de mots Bao et al. (2019) nous pourrons améliorer cette approche. Pourtant, une chose est certaine : les performances peuvent être considérablement améliorées en tenant compte de données labélisées. Par conséquent, une approche de type *few-shot learning* suivant une stratégie d'augmentation des données par le biais des réseaux de neurones type « Siamois » ou autres est largement recommandée pour traiter des problématiques semblables. Il est également possible d'utiliser des générateurs de phrases, e.g. GPT-2 à partir des mots-clés caractérisant les classes. Des approches dîtes N-way K-shot peuvent aussi être appliquées ; elles consistent à entraîner un modèle épisodiquement tel que à chaque épisode, l'entraînement est réalisé sur un sous-ensemble des données d'entraînement tiré aléatoirement tout en exploitant les données restantes pour calculer une représentation des classes, Snell et al. (2017), Vinyals et al. (2016).

Des approches d'augmentation basées sur la recherche de classement par ordre de pertinence entre les requêtes sont aussi exploitables, e.g. Triantafillou et al. (2017). On peut également imaginer un processus d'augmentation de données basé sur le principe d'une classification hiérarchique, en rajoutant des meta-classes regroupant les requêtes similaires deux à deux puis par lot. Enfin, des approches récentes proposent de tirer parti d'approches d'*adversarial learning* comme stratégie d'augmentation des données pour le *few-shot learning*, Zakharov et al. (2019).

5 Conclusion

Les problématiques de classification de textes courts sans données de supervision sont fréquentes dans des contextes industriels. Bien que des approches éprouvées de l'état de l'art permettent d'aborder la tâche en contexte non supervisé (*cold start*), notamment à l'aide d'approches à base de TF-IDF ou de techniques de plongements plus récentes, cette note souligne la pertinence d'étudier la considération de contextes faiblement supervisés peu contraignants en pratique (des dizaines de données labélisées peuvent suffire dans certains cas). Le passage vers des approches faiblement supervisées récentes (*few-shot* learning) puis, si possible, des approches d'apprentissage supervisé (en mode en ligne) permet en effet très souvent de garantir une nette augmentation de la qualité des systèmes de classification.

Références

Akbik, A., T. Bergmann, D. Blythe, K. Rasul, S. Schweter, et R. Vollgraf (2019). FLAIR : An easy-to-use framework for state-of-the-art NLP. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, Minneapolis, Minnesota, pp. 54–59. Association for Computational Linguistics.

Bao, Y., M. Wu, S. Chang, et R. Barzilay (2019). Few-shot text classification with distributional signatures. *arXiv preprint arXiv :1908.06039*.

Martin, L., B. Muller, P. J. Ortiz Suárez, Y. Dupont, L. Romary, É. Villemonte de la Clergerie, D. Seddah, et B. Sagot (2019). CamemBERT : a Tasty French Language Model. *arXiv e-prints*, arXiv :1911.03894.

Snell, J., K. Swersky, et R. Zemel (2017). Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pp. 4077–4087.

Triantafillou, E., R. Zemel, et R. Urtasun (2017). Few-shot learning through an information retrieval lens. In *Advances in Neural Information Processing Systems*, pp. 2255–2265.

Vinyals, O., C. Blundell, T. Lillicrap, D. Wierstra, et al. (2016). Matching networks for one shot learning. In *Advances in neural information processing systems*, pp. 3630–3638.

Zakharov, E., A. Shysheya, E. Burkov, et V. Lempitsky (2019). Few-shot adversarial learning of realistic neural talking head models. *arXiv preprint arXiv :1905.08233*.

Summary

In this note, we discuss different approaches of classification, without any supervision, of queries expressed on natural language into few classes given a textual description of each. We present then we compare approaches based on Information Retrieval as well as those based on word embeddings. Given few data of supervision, we show how can we process following a few-shot learning approach for further improvement of our classification using few annotated data then by exploring the data augmentation strategy with classes descriptions.