Détectez les bad buzz grâce au Deep Learning

Le traitement automatique du langage naturel ou Natural Language Processing (NLP) en anglais, est un domaine impliquant la linguistique, l'informatique et l'intelligence artificielle dans le but de créer des outils de traitement de la langue naturelle pour diverses applications. Un des applications est l'analyse de sentiment. Le but est de déterminer si le sentiment dégagé par une phrase est positif ou négatif.

La compagnie aérienne, Air Paradis, souhaite anticiper les bad buzz sur les réseaux sociaux. Pour ce faire, il est intéressant de tester plusieurs approches. Une approche « API sur étagère » en utilisant l'API du service cognitif proposé par Microsoft Azure, une approche « Modèle sur mesure simple » en utilisant une interface graphique d'un service Azure Machine Learning Studio et une approche « Modèle sur mesure avancé » basé sur des réseaux de neurones seront testées et comparées.

1. API sur étagère

La première approche testée est l'API sur étagère. Microsoft Azure fournit des fonctionnalités basées sur le cloud du traitement du langage naturel : l'exploration de texte, l'analyse de texte, l'analyse des sentiments, l'exploration des avis, l'extraction d'expressions clés, la détection de la langue, la reconnaissance d'entités nommées. Dans le cas de cette approche, l'API sera sollicité pour l'analyse des sentiments. Dans un premier temps, il faut créer une ressource Azure Cognitive Services (Service Language). Une fois la ressource déployée, il faut récupérer la clé et le point de terminaison qui sont ensuite stockés dans une variable d'environnement. Ainsi, on peut authentifier le client et solliciter l'API dans le but d'analyser les sentiments.

L'API retourne un sentiment détecté parmi « mixed », « positive », « negative » et « neutral » et trois scores « positive », « neutral », « negative ». Le retour de l'API devra donc être reclassifier en deux sentiments « positive » et « negative ». Par exemple, une régression logistique permettra de classifier cette sortie.

L'API présente un temps d'exécution de 26s pour traiter 1600 tweets et le score (recall) obtenu est de 0,83. Ainsi, cette approche présente un temps d'exécution faible et un très bon score, sans appliquer un prétraitement sur les données textuelles. Cette approche présente quand même désavantage c'est qu'elle présente un coût pour une utilisation à long terme.

2. Modèle sur mesure simple

Azure Machine Learning Studio est un environnement de développement intégré basé sur une interface graphique pour la construction et l'opérationnalisation du workflow Machine Learning sur Azure. Cet outil sera donc testé pour l'approche « Modèle sur mesure simple ».

L'interface drag & drop est très pratique pour mettre en place un modèle pour l'analyse de sentiment. Une fois les jeux de données chargés dans l'environnement, les colonnes pertinentes peuvent être sélectionnées pour extraire les n-gram. Il suffit d'entrainer, par exemple, une régression logistique sur le jeu d'entrainement et évaluer le modèle sur le jeu de test.

En ayant appliqué un prétraitement sur les données textuelles, le temps d'exécution de cette approche est d'environ 1 minute et le score (recall) est de 0 ,77. Le coût de ce service est plus faible que l'approche « API sur étagère ».

3. Modèle sur mesure avancé

Pour effectuer une analyse de sentiments, la troisième approche à tester est un modèle basé sur des réseaux de neurones. Trois modèles, deux approches de prétraitement et deux approches de word embedding seront comparés. Les modèles à comparer sont un modèle Keras de base avec embedding, un modèle Keras avec embedding et couche LSTM et un modèle BERT. Les deux approches de prétraitement sont la lemmatisation et le stemming et enfin les méthodes de prolongements de mots sont Word2Vec et GloVe.

Pour présenter rapidement la méthode Word2Vec, c'est un réseau de neurones superficiels à deux couches qui est formé pour reconstruire les contextes linguistiques des mots. Pendant l'entraînement, il va ajuster les poids pour réduire une fonction de perte et ce sont ces poids permettront de calculer le prolongement de mots (word embedding). Les vecteurs de mots sont positionnés dans l'espace vectoriel de telle sorte que les mots qui partagent des contextes communs dans le corpus soient situés à proximité les uns des autres dans l'espace.

En ce qui concerne la méthode GloVe, c'est un algorithme d'apprentissage non supervisé permettant d'obtenir des représentations vectorielles de mots. L'entraînement est effectué sur des statistiques globales agrégées de co-occurrence mot-mot à partir d'un corpus, et les représentations résultantes présentent des sous-structures linéaires de l'espace vectoriel de mots.

BERT (Bidirectional Encoder Representations from Transformers) est un modèle de représentation de textes écrit en langage naturel. La représentation faite par BERT a la particularité d'être contextuelle. C'est-à-dire qu'un mot n'est pas représenté de façon statique comme dans un embedding classique, mais en fonction du sens du mot dans le contexte du texte. Par exemple, le mot « baguette » aura des représentations différentes dans « la baguette du magicien » et « la baguette du boulanger ». En plus, le contexte de BERT est bidirectionel, c'est-à-dire que la représentation d'un mot fait intervenir à la fois les mots qui le précèdent et les mots qui le suivent dans une phrase.

Le comparatif des différentes approches se trouve dans le tableau ci-dessous :

Approche testé	**Prétraitement**	**Word Embedding**	**Recall**	**Temps d'exécution**	**Coût estimé**
API sur étagère			0,82	00:00:26	+++
Modèle sur mesure simple	Lemmatisation		0,76	00:01:00	+
	Stemming		0,77	00:01:00	
Modèle keras de base avec embedding	Lemmatisation	Word2Vec	0,74	00:02:18	+
		GloVe	0,65	00:02:01	
	Stemming	Word2Vec	0,67	00:02:18	
		GloVe	0,62	00:02:01	
Modèle keras de base avec embedding et couche LSTM	Lemmatisation	Word2Vec	0,77	00:55:55	++
		GloVe	0,75	00:49:23	
	Stemming	Word2Vec	0,70	00:47:55	
		GloVe	0,73	00:53:44	
Modèle Bert	Lemmatisation		0,91	00:56:07	++
	Stemming		0,89	00:56:02	