

# Rapport de projet

---

ANALYSE AUTOMATIQUE D'AVIS CLIENTS

Lionel Jospin KENNE NZANGEM & Karl Sondeji  
MSC 2 DATA ENGINEERING & CLOUD COMPUTING | AIVANCITY

# Analyse automatique d'avis clients

## *Aspect-Based Sentiment Analysis appliqué aux avis Amazon, catégorie Baby Products*

Advanced Practical Machine Learning, Professeur Souheil Hanoune, aivancity School of AI and Data

Lionel

Jospin KENNE NZANGEM & Karl Sondeji — MSc 2 Data Engineering & Cloud Computing (2025–2026)

Rapport de projet

## 1. Introduction et problématique

Les entreprises reçoivent chaque jour un volume important d'avis clients qu'il est difficile d'analyser manuellement. Une simple note globale ne permet pas d'identifier précisément les points forts et les axes d'amélioration d'un produit ou d'un service. Un même avis peut être très positif sur le prix et très négatif sur la sécurité, sans que la note moyenne ne le révèle. Ce projet a pour objectif de développer un système de traitement automatique du langage capable d'extraire, pour chaque avis, les aspects concrets évoqués (qualité du produit, prix, sécurité, facilité d'usage, etc.) et le sentiment associé à chacun d'eux. Cette tâche est connue sous le nom d'Aspect-Based Sentiment Analysis, ou ABSA.

Le domaine retenu est celui des produits pour bébés, à partir du corpus public Amazon Reviews 2023, catégorie Baby Products. Ce choix correspond à un cas d'usage directement transposable à des enseignes de puériculture, des plateformes e-commerce ou des services qualité, qui ont besoin de comprendre finement les retours clients au-delà d'une note globale.

## 2. Données et méthodologie de labellisation

Le jeu de données brut est constitué d'avis clients réels (texte, note, statut d'achat vérifié) issus d'Amazon Reviews 2023. Aucune annotation manuelle d'aspect ni de sentiment n'étant disponible, une étape de pseudo labellisation par weak supervision a été mise en place afin de transformer ces avis bruts en un jeu de données exploitable pour l'entraînement supervisé des modèles.

### 2.1 Pipeline de pseudo labellisation

Chaque avis est d'abord segmenté en phrases avec spaCy (module preprocess.py). Chaque phrase suit ensuite deux étapes de classification automatique, assurées par des modèles Hugging Face pré entraînés.

- Détection d'aspect. Un modèle zero shot NLI (DistilBERT MNLI, ou BART large MNLI pour une meilleure qualité) évalue la phrase contre dix catégories d'aspects candidates : qualité du produit, prix, design, taille, confort, durabilité, sécurité, facilité d'usage, livraison, service client. Ces catégories sont formulées comme des hypothèses NLI. Le score obtenu est renforcé par des règles lexicales fondées sur des expressions régulières, qui rattrapent les aspects évidents ratés par le modèle zero shot (module aspects.py).
- Détection de sentiment. Un classifieur DistilBERT fine tuné sur SST-2 attribue une polarité binaire, positive ou négative, à chaque phrase (module classifiers.py).

L'orchestration complète du pipeline (pipeline.py, build\_aspect\_labels.py) traite les avis par lots, avec reprise automatique en cas d'interruption, ce qui est important compte tenu du volume de données et du temps de calcul nécessaire sur CPU.

### 2.2 Préparation et répartition des données

Le nettoyage (prep\_data.py) décode les entités HTML résiduelles, retire les balises et les URLs, normalise la ponctuation typographique et déduplique les lignes identiques. Le découpage en ensembles d'entraînement, de validation et de test est réalisé par groupe sur l'identifiant de l'avis (review\_idx), avec un GroupShuffleSplit dans les proportions 80, 10 et 10 pour cent. Cette méthode garantit qu'un même avis, pouvant contenir plusieurs phrases, ne se retrouve jamais réparti entre plusieurs ensembles, ce qui éviterait une fuite de données. Un vocabulaire est ensuite construit exclusivement sur l'ensemble d'entraînement, avec une

fréquence minimale de deux occurrences et une taille maximale de 20000 tokens, pour la tokenisation du modèle Bi-LSTM. Le modèle BERT utilise directement son propre tokenizer WordPiece sur le texte nettoyé, sans passer par ce vocabulaire.

Le jeu de test final comprend 5569 exemples, sous forme de paires phrase et aspect, répartis en 3580 exemples positifs et 1989 exemples négatifs. Cette distribution déséquilibrée a un impact sur les résultats, discuté en section 5.

### 3. Formulation de la tâche et architectures des modèles

Les deux modèles développés résolvent la même tâche formelle : étant donnée une paire composée d'un aspect et d'une phrase, prédire un sentiment binaire, positif ou négatif. L'aspect fait partie intégrante de l'entrée du modèle. Sans lui, il serait impossible de distinguer, dans la phrase *the price is great but the safety strap is flimsy*, que l'aspect *price* est positif alors que l'aspect *safety* est négatif dans la même phrase. Deux architectures ont été entraînées et comparées : un réseau récurrent entraîné entièrement depuis zéro, et un modèle de type transformeur pré entraîné, adapté par fine tuning.

#### 3.1 Baseline Bi-LSTM (train\_baseline.py)

Le modèle de référence comporte les couches suivantes, dans l'ordre du calcul.

- Une couche d'embedding de mots, de dimension 128, initialisée aléatoirement et apprise pendant l'entraînement, avec un indice de padding fixé à zéro. La taille de cette matrice dépend du vocabulaire construit sur l'ensemble d'entraînement, plafonné à 20000 tokens (soit au maximum environ 2,56 millions de paramètres pour cette seule couche).
- Une couche d'embedding d'aspect, de dimension 32, avec un vecteur distinct pour chacune des dix catégories d'aspects (320 paramètres au total).
- Un réseau récurrent LSTM bidirectionnel à une seule couche, avec 128 unités cachées par direction. La séquence de mots est parcourue simultanément de gauche à droite et de droite à gauche. Le dernier état caché de chaque direction est concaténé, ce qui produit une représentation de phrase de dimension 256. Ce bloc récurrent compte environ 264000 paramètres entraînés.
- Une concaténation entre la représentation de phrase (256) et l'embedding de l'aspect cible (32), formant un vecteur de dimension 288.
- Une tête de classification à deux couches linéaires : une première couche cachée de 288 vers 128 neurones, suivie d'une activation ReLU et d'un dropout à 0,3, puis une seconde couche linéaire de 128 vers 2 neurones, correspondant aux deux classes de sortie. Cette tête représente environ 37250 paramètres.

Hors la matrice d'embedding de mots, dont la taille varie avec le vocabulaire, le reste du réseau représente environ 302000 paramètres entraînés. L'entraînement utilise l'optimiseur Adam avec un taux d'apprentissage de  $1e-3$ , des lots de 32 exemples, jusqu'à 30 époques, avec un arrêt anticipé si le F1 macro de validation ne progresse plus pendant 5 époques consécutives.

#### 3.2 BERT fine tuné (train\_bert.py)

Le second modèle repose sur bert-base-uncased, un transformeur pré entraîné composé de 12 couches d'encodeur empilées. Chaque couche applique un mécanisme d'auto-attention multi-tête, avec 12 têtes d'attention, sur une représentation cachée de dimension 768, suivi d'un réseau de neurones à propagation avant de dimension interne 3072 (soit quatre fois la taille cachée), avant de revenir à la dimension 768 en sortie de couche. L'ensemble du modèle pré entraîné compte environ 110 millions de paramètres, tous ajustés pendant le fine tuning, sans gel de couche.

La phrase et l'aspect sont encodés conjointement, au format dit BERT-SPC : la séquence d'entrée est construite comme le jeton spécial CLS, suivi des tokens de l'aspect, du jeton séparateur SEP, des tokens de la phrase, puis d'un second jeton SEP. Cette mise en forme permet au mécanisme d'attention de mettre

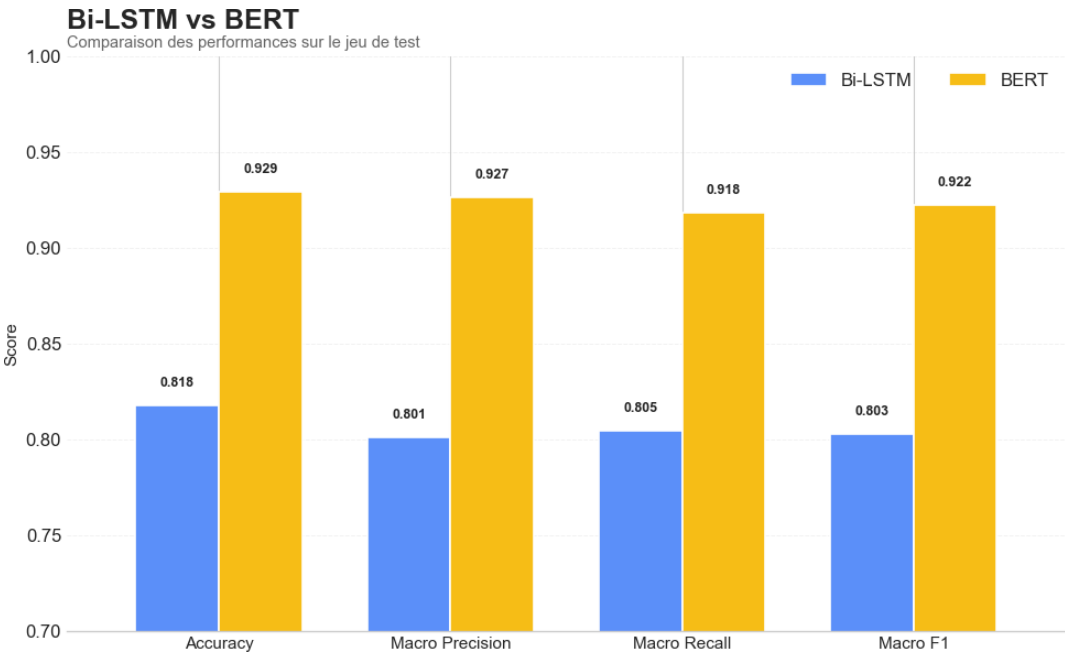
directement en relation les mots de la phrase avec l'aspect ciblé dès la première couche, ce qui constitue une différence structurelle majeure avec le Bi-LSTM, qui encode la phrase et l'aspect séparément avant de les combiner tardivement. La longueur maximale de séquence est fixée à 96 tokens.

La tête de classification ajoutée au-dessus du transformeur reprend l'architecture standard de bert-base pour la classification de séquences : la représentation du jeton CLS en sortie de la douzième couche est d'abord passée dans une couche linéaire de pooling (768 vers 768) suivie d'une activation tangente hyperbolique, puis dans un dropout, avant une dernière couche linéaire de 768 vers 2 neurones produisant les scores des deux classes. L'entraînement utilise l'optimiseur AdamW avec un taux d'apprentissage de 2e-5, une décroissance de poids de 0,01, un warmup linéaire sur 10 pour cent des pas, des lots de 16 exemples, jusqu'à 4 époques, avec un arrêt anticipé après 2 époques sans progression du F1 macro de validation.

#### 4. Résultats quantitatifs

Les deux modèles ont été évalués sur le même ensemble de test, 5569 exemples, avec le meilleur point de sauvegarde de chacun sélectionné selon le F1 macro de validation. Le F1 macro est retenu comme métrique de référence plutôt que l'accuracy seule, car il donne un poids égal aux deux classes malgré leur déséquilibre, 64 pour cent d'exemples positifs contre 36 pour cent de négatifs.

| Métrique        | Baseline Bi-LSTM | BERT fine tuné | Gain BERT  |
|-----------------|------------------|----------------|------------|
| Accuracy        | 81,79 %          | 92,93 %        | +11,13 pts |
| Precision macro | 80,13 %          | 92,66 %        | +12,53 pts |
| Recall macro    | 80,46 %          | 91,84 %        | +11,37 pts |
| F1-score macro  | 80,29 %          | 92,22 %        | +11,94 pts |



| Classe            | Precision Bi-LSTM | Recall Bi-LSTM | F1 Bi-LSTM | Precision BERT | Recall BERT | F1 BERT |
|-------------------|-------------------|----------------|------------|----------------|-------------|---------|
| negative (n=1989) | 0,74              | 0,76           | 0,75       | 0,92           | 0,88        | 0,90    |
| positive (n=3580) | 0,86              | 0,85           | 0,86       | 0,94           | 0,96        | 0,95    |

BERT surpasse la baseline sur toutes les métriques, avec un gain particulièrement marqué sur la classe négative, minoritaire, dont le F1 passe de 0,75 à 0,90. Ce résultat est cohérent avec l'apport attendu du pré entraînement à grande échelle pour mieux généraliser sur une classe sous représentée. Sur les 5569 exemples de test, les deux modèles sont en désaccord sur 918 exemples, soit 16,5 pour cent. BERT corrige une erreur de la baseline dans 769 cas, soit 13,8 pour cent du total, mais introduit une nouvelle erreur là où la baseline avait raison dans 149 cas, soit 2,7 pour cent. Ces deux catégories de désaccord, ainsi que les cas où les deux modèles se trompent, 245 cas soit 4,4 pour cent, sont analysées qualitativement en section 5.

## 5. Analyse d'erreurs qualitative

Au-delà des métriques agrégées, l'examen d'exemples individuels tirés du fichier `categorized_errors.csv` permet de comprendre les situations où les prédictions des deux modèles divergent, et d'observer comment cela se relie au contexte de chaque phrase et au pipeline de pseudo labellisation décrit en section 2.1. Trois catégories sont examinées : les cas où BERT s'aligne avec le label alors que la baseline s'en écarte, les cas inverses, et les cas où les deux modèles s'écartent du label.

### 5.1 BERT corrige la baseline

| Phrase                                                                                                                               | Aspect               | Label    | Bi-LSTM  | BERT     |
|--------------------------------------------------------------------------------------------------------------------------------------|----------------------|----------|----------|----------|
| these locks really helped in securing them, they did not open even when kid tried forcedly, 1 star less due to difficulty in removal | qualité du produit   | positive | negative | positive |
| This is also easy to put on the baby, it zips right up or down, with the two way zipper.                                             | facilité d'usage     | positive | negative | positive |
| Nice but to small for my use.                                                                                                        | taille et ajustement | positive | negative | positive |

Le premier exemple est une phrase longue et globalement positive qui contient pourtant une négation (did not open) et se termine sur une remarque négative mineure (1 star less). La baseline semble s'être laissée dominer par ce signal de fin de phrase et par la négation, alors que BERT intègre correctement l'ensemble du contexte pour dégager le sentiment global positif dominant. Le deuxième exemple est plus surprenant, car aucune négation n'est présente et la baseline échoue tout de même. Ceci suggère que ses embeddings, appris depuis zéro sur un vocabulaire limité, représentent mal certaines tournures que BERT, pré entraîné sur un corpus massif, traite nativement. Le troisième exemple illustre une particularité propre à l'analyse au niveau de la phrase isolée. Le texte est objectivement critique sur la taille, et la prédiction négative de la baseline correspond à une lecture humaine plausible à partir de cette seule phrase, tandis que la prédiction positive de BERT s'aligne avec le label du jeu de données. Les deux lectures restent défendables sans le contexte complet de l'avis d'origine.

### 5.2 BERT introduit une nouvelle erreur

| Phrase                                                                                             | Aspect              | Label    | Bi-LSTM  | BERT     |
|----------------------------------------------------------------------------------------------------|---------------------|----------|----------|----------|
| When she comes to visit, she picks this toy out of the pile of toys to play with first every time. | qualité du produit  | negative | negative | positive |
| However, with these locks he has not been able to get around it, so we are beyond impressed!       | design et apparence | negative | negative | positive |
| It was a little expensive and I hope it holds for a few years.                                     | qualité du produit  | positive | positive | negative |

Les deux premiers cas sont, comme en section 5.1, des exemples où la lecture au niveau de la phrase isolée peut s'écarter du label du jeu de données. La première phrase décrit un jouet préféré d'un enfant, un signal très positif pris isolément, mais porte un label negative, probablement établi à partir d'un passage différent ou plus large de l'avis d'origine. La seconde combine une double négation, has not been able to get around it, qui peut

se lire comme indiquant que le dispositif de sécurité fonctionne bien, renforcée par *beyond impressed*. Dans les deux cas, BERT prédit positive, une lecture cohérente avec le contenu de la phrase isolée, tandis que la baseline s'aligne avec le label. Le troisième exemple est un cas où BERT s'écarte du label alors que la baseline s'y aligne. Le langage y est hésitant plutôt que négatif, *a little expensive*, *I hope*. Ce type de formulation, où une nuance positive et prudente côtoie un mot à connotation négative, illustre la difficulté générale du langage nuancé pour l'analyse de sentiment automatique, y compris pour des modèles pré entraînés à grande échelle.

### 5.3 Les deux modèles se trompent

| Phrase                                              | Aspect             | Label    | Bi-LSTM  | BERT     |
|-----------------------------------------------------|--------------------|----------|----------|----------|
| This is a great sleeper sack!                       | confort et texture | negative | positive | positive |
| VIDEOID, This is a take anywhere light stroller.    | sécurité           | positive | negative | negative |
| It kept him safe and he could crawl out when ready. | sécurité           | negative | positive | positive |

Le premier exemple est le cas le plus net, dans cette analyse, où la phrase isolée et le label du jeu de données divergent nettement. La phrase est sans ambiguïté enthousiaste, *great*, et les deux modèles la lisent, de manière cohérente avec son contenu, comme positive. Le deuxième exemple révèle une limite de prétraitement. Un jeton résiduel de type identifiant de vidéo, artefact de balisage Amazon non nettoyé en amont, pollue une phrase par ailleurs neutre et sans signal de sentiment clair, ce qui influence les deux modèles de façon similaire. C'est une piste d'amélioration concrète du pipeline de nettoyage, module `preprocess.py`, fonction `clean_text`. Le troisième exemple est un cas d'ambiguïté réelle. *Kept him safe* est positif, mais *could crawl out when ready* peut aussi bien décrire une fonctionnalité voulue qu'une préoccupation de sécurité selon le contexte plus large de l'avis, non disponible au niveau de la phrase isolée.

## 6. Discussion critique et limites

L'analyse qualitative de la section 5 met en évidence une caractéristique importante de la méthodologie du projet, complémentaire au simple écart de performance entre les deux architectures. Une part des désaccords entre les modèles, et une partie des cas où les deux s'écartent du label, semble liée non pas à la qualité des architectures elles-mêmes, mais à la nature de la tâche de pseudo labellisation par weak supervision, qui analyse chaque phrase indépendamment plutôt que dans le contexte complet de l'avis d'origine. Les labels d'entraînement et de test sont générés automatiquement par un pipeline combinant une classification zero shot d'aspect et une classification de sentiment, une approche courante pour constituer un jeu de données ABSA à grande échelle en l'absence de budget d'annotation manuelle. Cette approche produit des labels dont la lecture au niveau de la phrase isolée peut occasionnellement différer de celle qu'un humain ferait en disposant du contexte complet de l'avis, ce qui explique une partie de l'écart mesuré entre les deux modèles sans remettre en cause la qualité de l'un ou l'autre. Dès le début du projet, une vérification manuelle a été réalisée sur une cinquantaine d'observations sélectionnées aléatoirement, afin de contrôler si les couples phrase-aspect correspondaient à la réalité selon une interprétation humaine. Par exemple, dans la phrase « *super produit mais un peu cher au vu des caractéristiques* », l'aspect *délai de livraison* n'est pas présent. Cette vérification a permis de constater que les prédictions n'étaient pas toujours exactes pour certaines phrases. Le seuil de confiance associé à la création des couples phrase-aspect a alors été relevé à 0,7 afin d'améliorer la fiabilité du pipeline.

Plusieurs limites secondaires, plus techniques, ont également été identifiées. Les hyperparamètres utilisés pour les deux modèles correspondent à des valeurs couramment recommandées dans la littérature (taux d'apprentissage de  $2e-5$  pour le fine tuning de BERT, architecture Bi-LSTM à une couche de 128 unités) plutôt qu'au résultat d'une recherche systématique par grille ou par validation croisée. Ce choix est courant pour un projet de ce format, mais une exploration plus large de l'espace des hyperparamètres, ou une comparaison avec d'autres modèles pré entraînés comme RoBERTa ou DistilBERT, pourrait affiner les

résultats obtenus. De la même manière, l'évaluation repose sur un seul découpage train, validation, test plutôt que sur une validation croisée à k plis, ce qui ne permet pas d'estimer la variance des performances mesurées. Le périmètre du projet est également restreint à une seule catégorie de produits, les articles pour bébés, et à une classification binaire du sentiment : la généralisation à d'autres catégories de produits ou la prise en compte d'une classe neutre n'ont pas été testées et constituent des extensions naturelles du travail. La segmentation de phrases par spaCy peut par ailleurs tronquer des phrases complexes au niveau de points de suspension, produisant des fragments grammaticalement incomplets. Le nettoyage de texte ne retire pas systématiquement certains artefacts de balisage Amazon, comme les jetons d'identifiant de vidéo, qui polluent le signal textuel pour les deux modèles de façon comparable. Enfin, l'analyse confirme que le taux d'erreur de BERT est plus élevé sur les phrases contenant un marqueur de négation ou de contraste, 9,77 pour cent contre 6,19 pour cent sans négation, un motif largement documenté dans la littérature ABSA où la négation reste une source de complexité même pour les architectures pré entraînées les plus récentes.

## 7. Conclusion

Ce projet démontre la faisabilité d'un système d'analyse d'avis clients par aspect entièrement construit à partir de données réelles non annotées, en combinant un pipeline de pseudo labellisation par weak supervision avec deux architectures PyTorch entraînées et comparées de bout en bout, un Bi-LSTM de référence et un BERT fine tuné. Le gain de performance de BERT, 11,9 points de F1 macro, confirme l'intérêt du transfer learning pour cette tâche, en particulier sur la classe négative minoritaire. L'analyse d'erreurs qualitative, au delà de confirmer ce résultat quantitatif, a mis en lumière le rôle du contexte de la phrase dans la pseudo labellisation, une piste d'amélioration intéressante pour une itération future du projet, en complément de l'ajustement des architectures. La démonstration Streamlit permet de rendre ce système accessible à un utilisateur final, enseigne de puériculture, plateforme e-commerce ou service qualité, qui peut soumettre un avis ou un corpus d'avis et visualiser immédiatement les aspects détectés et leur polarité.

## Références

- Devlin, J., Chang, M. W., Lee, K., Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.
- Pontiki, M., Galanis, D., Papageorgiou, H., Androutsopoulos, I., Manandhar, S., Mohammad, A. S., et al. (2016). SemEval-2016 Task 5: Aspect Based Sentiment Analysis.
- Hochreiter, S., Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8), 1735-1780.
- Lewis, M., et al. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension (modèle bart-large-mnli, zero shot NLI).