# DACE pour la désambiguïsation d'acronymes ferroviaires

El Mokhtar Hribach*†, Oussama Mechhour**†
Mohammed Elmonstaser***†, Yassine El Boudouri****†, Othmane Kabal*†

* Nantes Université, LS2N, Nantes 44300, France
el-mokhtar.hribach@univ-nantes.fr, othmane.kabal@univ-nantes.fr
** CIRAD, F-34398 Montpellier, France
oussama.mechhour@cirad.fr
*** Groupe SII, France
mohammed.elmontaser@sii.fr
**** Univ. Lille, CNRS, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France
yassine.el-boudouri@univ-lille.fr

**Abstract.** La désambiguïsation d'acronymes (DA) est un défi fondamental dans le traitement de textes techniques, en particulier dans les secteurs spécialisés où une forte ambiguïté complique l'analyse automatique. Cet article aborde la DA dans le contexte de la compétition TextMine'26 portant sur la documentation ferroviaire française. Nous présentons DACE (**D**ynamic Prompting, Retrieval **A**ugmented Generation, **C**ontextual Selection, and **E**nsemble Aggregation), un cadre méthodologique qui améliore les grands modèles de langage (LLM) grâce à un apprentissage en contexte adaptatif et l'injection de connaissances externes. En ajustant dynamiquement les prompts selon l'ambiguïté des acronymes et en agrégeant les prédictions d'ensemble, DACE atténue les hallucinations et gère efficacement les scénarios à faibles ressources. Notre approche a obtenu la première place de la compétition avec un score F1 de 0,9069.

## 1 Introduction

Ce travail s'inscrit dans le cadre de la compétition TextMine'26 (Luce Lefeuvre, 2025), organisée par l'association *Extraction et Gestion des Connaissances* (EGC). Cette compétition confronte l'état de l'art scientifique aux défis industriels de la fouille de textes. L'édition 2026, proposée par la SNCF (Société Nationale des Chemins de fer Français), porte sur le traitement automatique des acronymes.

Ce contexte industriel souligne l'importance générale de la désambiguïsation d'acronymes (DA), problème central du traitement automatique du langage et de l'extraction d'informations. Les acronymes sont omniprésents dans les domaines techniques, où ils désignent de manière compacte des processus, entités, équipements ou composants logiciels. Cette concision entraîne toutefois une forte ambiguïté : de nombreux acronymes sont hautement polysémiques

†Tous les auteurs ont contribué de manière égale à ce travail.

(Zahariev, 2004), et leur interprétation dépend du domaine et du contexte local. Dans des secteurs critiques et réglementés comme le ferroviaire, lever cette ambiguïté est indispensable pour garantir une communication fiable, le partage des connaissances et l'automatisation des tâches en aval (maintenance, conformité, recherche technique) (Kong and Ahn, 2024).

Les approches de DA ont évolué des heuristiques à base de règles (Schwartz and Hearst, 2002) vers des modèles supervisés utilisant des caractéristiques manuelles (Okazaki and Ananiadou, 2006), puis vers des méthodes fondées sur des représentations contextuelles et des transformers spécialisés (Li et al., 2015; Veyseh et al., 2020). Si les modèles supervisés atteignent de bonnes performances en présence d'annotations abondantes, ils généralisent mal en contexte peu doté ou lors d'un changement de domaine. Les grands modèles de langage (LLM) proposent une alternative complémentaire : grâce au prompt engineering et à l'apprentissage en contexte, ils permettent une désambiguïsation efficace avec peu, voire sans données annotées (Kugic et al., 2024). Cette flexibilité soulève toutefois des questions sur la conception des prompts, le choix des exemples et l'intégration de connaissances structurées pour garantir la fiabilité.

Nous présentons dans cet article **DACE** (**D**ynamic Prompting, Retrieval-**A**ugmented Generation, **C**ontextual Selection et **E**nsemble Aggregation), un cadre opérationnel de désambiguïsation d'acronymes pour les textes techniques ferroviaires. DACE repose sur quatre composants complémentaires : le **prompting dynamique**, qui ajuste la complexité du prompt selon l'ambiguïté de l'acronyme ; la **RAG**, qui intègre des glossaires ferroviaires et des fragments d'ontologie ; la **sélection contextuelle**, qui fournit un ensemble réduit et équilibré d'exemples en contexte ; et l'**agrégation en ensemble**, qui combine plusieurs configurations de modèles pour renforcer la robustesse. L'ensemble vise à concilier efficacité en faible supervision, spécialisation métier et stabilité opérationnelle.

Nous évaluons DACE sur le corpus TextMine'26 et analysons ses performances sur des acronymes fréquents, rares et fortement polysémiques. Les résultats montrent que le prompting dynamique et la sélection contextuelle améliorent nettement la précision en régime peu doté, tandis que l'apport de connaissances externes et l'agrégation multi-modèles renforcent la cohérence et la résilience aux erreurs.

# 2 Travaux Connexes

## 2.1 Désambiguïsation d'Acronymes

La désambiguïsation d'acronymes est étudiée de longue date en traitement automatique du langage, initialement à travers des méthodes heuristiques et à base de règles fondées sur la correspondance de motifs et l'extraction entre parenthèses. Les travaux de Schwartz and Hearst (2002) ont notamment formalisé l'identification des formes longues à partir de dictionnaires et du contexte local, fournissant une approche non supervisée sans données annotées.

Les recherches ultérieures ont formulé la résolution d'acronymes comme un problème de classification supervisée ou une variante de la désambiguïsation du sens des mots (Chen et al., 2023). Les premiers systèmes reposaient sur des SVM, des classifieurs Naive Bayes et des caractéristiques linguistiques manuelles, généralement associées à des dictionnaires extraits de corpus annotés. L'introduction des plongements de mots a ensuite permis de projeter con-

texte et expansions candidates dans un espace sémantique commun, favorisant la similarité distributionnelle plutôt que la correspondance de surface (Wu et al., 2015; Song et al., 2022).

Les modèles Transformers ont encore amélioré l'état de l'art. À la suite du benchmark SciAD, des variantes de BERT telles que SciBERT ont établi de nouveaux résultats de référence sur des jeux de données d'acronymes scientifiques (Pan et al., 2021; Chen et al., 2023). La question de la généralisation inter-domaines a également émergé, avec des ressources comme MadDog (Veyseh et al., 2021) et GLADIS (Chen et al., 2023) élargissant la couverture à des domaines hétérogènes et à grande échelle.

Malgré ces progrès, le changement de domaine et les contextes à faibles ressources demeurent difficiles. Les modèles supervisés se dégradent lorsque les distributions d'entraînement et de test divergent, en particulier dans des domaines spécialisés. Les approches non ou faiblement supervisées exploitent des données non étiquetées ou la cohérence documentaire, mais restent généralement en retrait des méthodes pleinement supervisées (Song et al., 2022). Ces limites motivent des stratégies plus adaptatives, capables de généraliser avec une supervision minimale.

Les travaux récents explorent ainsi les grands modèles de langage (LLM) pour la désambiguïsation zero-shot ou few-shot. Kugic et al. (2024) montrent que GPT-4 obtient des performances compétitives sur des benchmarks anglais sans affinage, suggérant une connaissance acquise lors du pré-entraînement. Toutefois, les performances chutent lors de changements de domaine ou de langue, comme observé pour l'allemand et le portugais, avec une stabilité encore plus faible pour des modèles open-source tels que LLaMA 2 (Touvron et al., 2023). Les approches few-shot ou basées sur des données synthétiques apportent des gains limités (Kugic et al., 2025), sans résoudre pleinement l'ambiguïté dans des contextes hautement spécialisés.

## 2.2 Apprentissage en Contexte et Ingénierie de Prompt

L'apprentissage en contexte constitue une alternative flexible à l'affinage spécifique à la tâche, permettant aux LLM d'exécuter de nouvelles tâches à partir d'instructions ou d'exemples intégrés au prompt. L'ingénierie de prompt est ainsi devenue un axe de recherche majeur, la structure du prompt et la sélection des exemples ayant un impact significatif sur les performances (Sumanathilaka et al., 2025). Pour des tâches proches, comme la désambiguïsation du sens des mots, les LLM révèlent une connaissance linguistique latente, notamment lorsque la tâche est formulée comme une inférence en langage naturel (Sainz et al., 2023).

Le prompting zero-shot fournit souvent une base raisonnable, mais souffre d'instabilité et d'hallucinations dans les domaines spécialisés. Le prompting few-shot améliore généralement précision et cohérence en contraignant l'espace de sortie et en illustrant le raisonnement attendu (Min et al., 2022). Des travaux récents soulignent par ailleurs l'importance de la sélection des démonstrations : des exemples sémantiquement proches ou informatifs surpassent systématiquement l'échantillonnage aléatoire (Liu et al., 2022). En désambiguïsation d'acronymes, des démonstrations équilibrées ou représentant des sens rares réduisent le biais de fréquence et favorisent un raisonnement guidé par le contexte.

Le prompting augmenté par la récupération permet en outre de pallier les limites des connaissances paramétriques des LLM en injectant des informations externes à l'inférence. Dans les domaines techniques, l'incapacité à rappeler des entités de longue traîne ou une terminologie spécialisée conduit fréquemment à des expansions hallucinées (Kandpal et al., 2023).

Les mécanismes de récupération atténuent ce phénomène en fournissant des définitions et exemples issus de sources de référence. Le paradigme de la Retrieval-Augmented Generation (RAG) (Lewis et al., 2020) s'est montré efficace pour les tâches intensives en connaissances et se prête particulièrement à la désambiguïsation d'acronymes, en contraignant les prédictions aux candidats valides tout en améliorant l'alignement factuel et l'interprétabilité (Shuster et al., 2021).

# 3 Approche Proposée

## 3.1 Formulation du Problème

Nous définissons la désambiguïsation d'acronymes comme suit. Soit $a_i$ une occurrence d'acronyme dans un contexte $t_i$, associée à un ensemble de candidats $O_i = \{O_{i_1}, \ldots, O_{i_k}\}$. L'objectif est de prédire l'étiquette $y_i$, identifiant la ou les expansions correctes parmi $O_i$, qui peut contenir une, plusieurs ou aucune option correcte.

L'ensemble d'entraînement de $\nu$ exemples annotés est $E_{\text{train}} = \{((a_i, t_i, O_i), y_i)\}_{i=1}^{\nu}$, et l'ensemble de test de $\mu$ instances est $E_{\text{test}} = \{(a_j, t_j, O_j)\}_{j=1}^{\mu}$, avec étiquettes inconnues. Le test distingue les acronymes vus à l'entraînement ($C_{\text{test}\cap\text{train}}$) des acronymes inédits ($C_{\text{test}\setminus\text{train}}$).

Le système de désambiguïsation est formalisé comme une fonction de mappage :

$$f_{\text{AD}} : (a_j, t_j, O_j) \mapsto \hat{y}_j.$$

## 3.2 Vue d'ensemble du cadre DACE

S'appuyant sur la définition formelle de la tâche, nous introduisons **DACE**, un cadre modulaire conçu pour améliorer les performances des LLM grâce à un apprentissage en contexte structuré. L'architecture orchestre quatre composants synergiques : (1) **Prompting Dynamique** pour adapter la complexité des instructions ; (2) **Génération Augmentée par la Récupération** pour injecter des connaissances du domaine ; (3) **Sélection Contextuelle** pour fournir des démonstrations pertinentes ; et (4) **Agrégation d'Ensemble** pour stabiliser les prédictions. Une vue d'ensemble du pipeline complet est présentée dans la Figure 1. Les sous-sections suivantes détaillent l'implémentation de ces modules.

## 3.3 Prompting Dynamique

Pour gérer l'hétérogénéité des usages d'acronymes dans TextMine'26, nous remplaçons les instructions statiques par un constructeur de prompt dynamique. Ce module synthétise les connaissances récupérées (Section 3.5) et les démonstrations few-shot sélectionnées (Section 3.4), tout en adaptant la stratégie d'instruction au profil de difficulté de chaque instance.

L'analyse préliminaire a montré qu'un prompt « taille unique » échoue à généraliser sur acronymes fréquents et rares, notamment lorsque les candidats présentent un fort chevauchement lexical. Pour y remédier, une fonction de basculement choisit entre deux gabarits selon le statut d'entraînement et la similarité des candidats.

**Template A (Contexte Standard) :** appliqué par défaut si l'acronyme est connu ($a_j \in C_{\text{test}\cap\text{train}}$), où les exemples en contexte suffisent pour une désambiguïsation fiable.

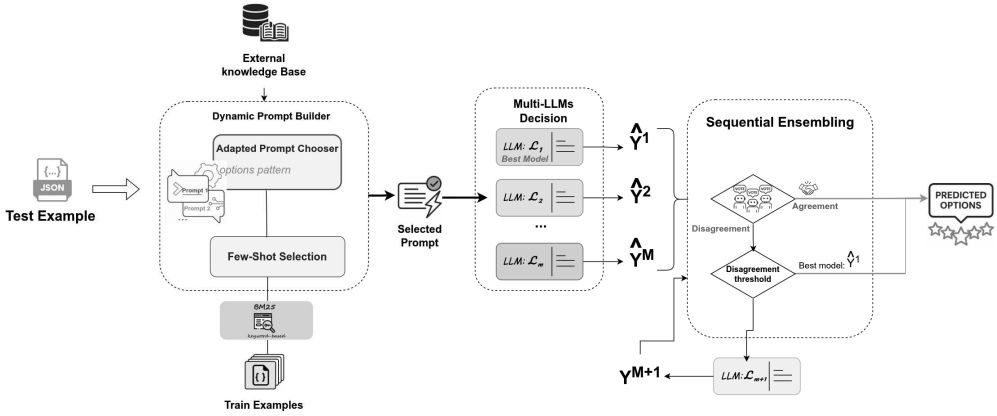



FIG. 1 – *Architecture du cadre DACE.*

**Template B (Focalisé sur la Désambiguïsation) :** déclenché uniquement si l'acronyme est inédit ($a_j \in C_{\text{test}\setminus\text{train}}$) et que l'ensemble de candidats $O_j$ contient des options fortement chevauchantes morphologiquement. Le chevauchement est mesuré par la similarité de Jaccard sur les radicaux (stemming). Ce gabarit impose un raisonnement explicite et privilégie les expansions techniques face aux termes génériques, réduisant les confusions entre candidats sémantiquement proches.

## 3.4 Sélection Contextuelle

Le module de Sélection Contextuelle s'active uniquement lorsque l'acronyme cible a été observé à l'entraînement ($a_j \in C_{\text{test}\cap\text{train}}$), correspondant au chemin few-shot (Template A). Pour ces instances, un index BM25 (Robertson et al., 2009) construit sur le corpus d'entraînement est utilisé afin de récupérer des occurrences contextuellement similaires.

Afin de maintenir un prompt informatif et non biaisé, la sélection repose sur deux mécanismes complémentaires :

1. **Échantillonnage équilibré**, garantissant une distribution égale d'exemples positifs (sens cible) et contrastifs, afin d'éviter un biais vers la classe majoritaire ;

2. **Déduplication sensible à la diversité**, qui élimine les exemples redondants via une similarité textuelle normalisée, maximisant la diversité sémantique dans une fenêtre contextuelle contrainte.

Pour une instance $(a_j, t_j)$, le module renvoie ainsi jusqu'à six démonstrations en contexte. Lorsque l'acronyme est connu, les exemples sont récupérés par BM25, puis filtrés par échantillonnage équilibré et déduplication, assurant des démonstrations à la fois pertinentes et complémentaires.

Pour les acronymes inédits ($a_j \in C_{\text{test}\setminus\text{train}}$), la sélection est désactivée et le système bascule en inférence zero-shot. Le choix du gabarit dépend alors uniquement de l'ambiguïté des candidats : le Template B est appliqué lorsque la similarité lexicale maximale entre expansions

dépasse un seuil fixé, signalant un fort chevauchement ; sinon, le Template A est conservé. Cette logique garantit que le contexte few-shot n'est utilisé que lorsque des exemples fiables existent, tandis que des instructions strictes sont réservées aux cas zero-shot les plus ambigus.

## 3.5 Génération Augmentée par la Récupération (Ancrage des Connaissances)

Afin de réduire l'écart sémantique entre les LLM généralistes et la terminologie ferroviaire spécialisée, nous avons construit une base de connaissances (BC) dédiée à la désambiguïsation d'acronymes. Celle-ci agrège trois sources complémentaires : des glossaires ferroviaires publics, la documentation technique open-source de la SNCF et des expansions validées extraites de l'ensemble d'entraînement. La BC est structurée comme un dictionnaire associant chaque acronyme à une liste d'expansions autorisées, injectées directement dans le prompt, constituant ainsi une ressource lexicale fiable et spécifique au domaine.

Pour exploiter cette ressource, nous implémentons un mécanisme de récupération hybride. Pour chaque instance de test, le système interroge la BC afin d'obtenir les formes longues candidates, leurs définitions validées et des exemples d'usage représentatifs. Cette récupération agit comme une couche d'ancrage, contraignant le raisonnement du modèle. En conditionnant l'inférence sur des connaissances explicites et vérifiées plutôt que sur la seule mémoire paramétrique, nous renforçons l'alignement des prédictions avec la sémantique technique et les contraintes opérationnelles du domaine ferroviaire.

## 3.6 Agrégation d'Ensemble

Le composant final du cadre DACE est le module d'Agrégation d'Ensemble. Pour atténuer la nature stochastique des LLM individuels et réduire les risques d'hallucination, nous employons une stratégie d'apprentissage d'ensemble. Cette approche utilise un pool diversifié de $M$ modèles candidats, filtrant et combinant systématiquement leurs sorties pour assurer une prise de décision robuste.

Une fois le modèle de prompt $T$ et (le cas échéant) l'ensemble few-shot $\mathcal{F}_j$ finalisés, le système instancie le prompt spécifique $P(e_j^{\text{test}})$ pour l'instance de test $e_j^{\text{test}} = (a_j, t_j, O_j)$. Ce prompt est diffusé à l'ensemble des modèles sélectionnés. Formellement, chaque LLM $\mathcal{L}_m$ (où $m \in \{1, \ldots, M\}$) mappe l'entrée vers un ensemble de prédictions :

$$\hat{Y}_j^{(m)} = \mathcal{L}_m\big(P(e_j^{\text{test}})\big),$$

où $\hat{Y}_j^{(m)} \subseteq \{1, \ldots, |O_j|\}$ désigne les indices des options prédites. Si un modèle échoue à identifier une option valide ou prédit "aucun", $\hat{Y}_j^{(m)} = \varnothing$. L'étape d'inférence produit ainsi une collection de prédictions $\hat{\mathcal{Y}}_j = \{\hat{Y}_j^{(1)}, \ldots, \hat{Y}_j^{(M)}\}$.

Plutôt qu'un vote majoritaire naïf sur tous les $M$ modèles, nous construisons la décision finale $\hat{y}_j$ en utilisant une logique en cascade qui donne la priorité aux modèles performants et complémentaires. Ce processus est régi par trois principes :

**1. Sélection de Sous-ensemble :** Nous limitons strictement le pool de vote à un sous-ensemble $\mathcal{S} \subseteq \{1, \ldots, M\}$ de modèles qui démontrent de fortes performances individuelles et une diversité d'erreurs sur l'ensemble de validation.

**2. Vote Majoritaire avec Départage :** Pour une instance donnée, nous calculons le consensus majoritaire sur $\mathcal{S}$.

- **Cas Standard :** Si $|\mathcal{S}|$ est impair, ou si $|\mathcal{S}|$ est pair mais qu'aucune égalité ne survient, la sortie finale est la prédiction majoritaire :

$$\hat{y}_j = \mathrm{MajVote}\big(\{\hat{Y}_j^{(m)} : m \in \mathcal{S}\}\big).$$

- **Départage :** Si $|\mathcal{S}|$ est pair et qu'une égalité parfaite survient (par exemple, une division 2 contre 2), nous interrogeons un *modèle de départage* désigné $\mathcal{L}_{m^\star}$ (où $m^\star \notin \mathcal{S}$). Ce modèle est sélectionné spécifiquement pour ses schémas de raisonnement complémentaires par rapport à $\mathcal{S}$. Le vote est ensuite recalculé sur l'ensemble étendu $\mathcal{S} \cup \{m^\star\}$ pour forcer une décision.

**3. Repli de Compétence :** Dans les rares cas de *forte divergence*, où l'ensemble échoue à atteindre un consensus significatif (c'est-à-dire plusieurs prédictions conflictuelles avec un faible soutien), le système écarte le vote et se replie sur le modèle unique ayant la plus haute précision historique, noté $m_{\text{best}}$ :

$$\hat{y}_j = \hat{Y}_j^{(m_{\text{best}})}.$$

# 4 Expérimentations

## 4.1 Jeu de données

Le jeu de données du défi TextMine'26 comprend deux partitions : un ensemble d'entraînement de 492 exemples annotés et un ensemble de test de 519 exemples. Chaque instance correspond à un court extrait de texte technique ferroviaire contenant un acronyme cible, la tâche consistant à sélectionner la forme étendue correcte parmi une liste de candidats fournie.

Le corpus couvre un ensemble diversifié d'acronymes présentant des degrés d'ambiguïté variables. La Figure 2 illustre le nombre d'acronymes uniques dans chaque partition et met en évidence un défi majeur : bien que certains acronymes soient partagés, 95 acronymes n'apparaissent que dans l'ensemble de test et sont absents de l'entraînement.

Le nombre d'expansions candidates par exemple varie fortement, de 2 à 13 options (Figure 3a), reflétant des niveaux d'ambiguïté hétérogènes, allant de décisions binaires à des choix parmi de larges ensembles de candidats.

Chaque instance peut admettre zéro, une ou plusieurs expansions correctes, correspondant à des scénarios réalistes de désambiguïsation. La Figure 3b présente la distribution des réponses correctes dans l'ensemble d'entraînement : 68 exemples (13,8 %) ne contiennent aucune option valide, tandis que 9 exemples acceptent plusieurs expansions, imposant aux systèmes de gérer l'abstention et l'ambiguïté contextuelle.

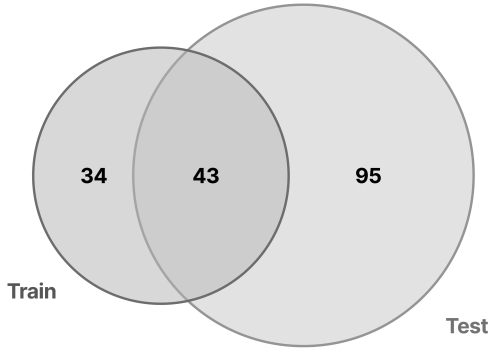



FIG. 2 – *Distribution des acronymes uniques entre les ensembles d'entraînement et de test.*

Dans l'ensemble, ces caractéristiques orientent la conception méthodologique. La présence d'acronymes rares ou uniquement observés en test limite les approches strictement supervisées, tandis que la variabilité des ensembles de candidats et les cas limites requièrent des stratégies d'inférence flexibles. Enfin, le caractère prescriptif et spécialisé de la documentation ferroviaire motive l'intégration de connaissances techniques externes et d'un prompting sensible au contexte.

## 4.2 Configuration Expérimentale

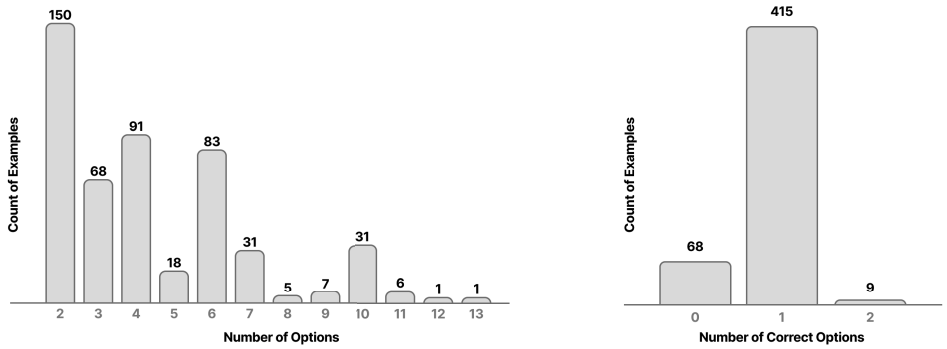
Toutes les expériences ont été conduites à l'aide des API officielles d'Anthropic et de Google. Sur la base d'une validation préliminaire sur l'ensemble d'entraînement, nous avons retenu trois LLM pour l'agrégation : **Claude 3.5 Sonnet (v2-20241022)**, **Claude 4.1 Opus (v4.1-20250805)** et **Gemini 2.5 Pro**.

Afin de garantir la reproductibilité et de limiter la variance stochastique, la température a été fixée à $\tau = 0$ (décodage glouton) pour toutes les inférences. Les sorties sont contraintes via le mode JSON, imposant la génération d'un dictionnaire associant chaque candidat à un verdict booléen (par ex. `{"A": true, "B": false}`). Ce format strict élimine les erreurs de parsing et concentre le modèle sur la classification binaire des options.

L'ensemble du code, incluant les gabarits de prompts et les configurations expérimentales, est mis à disposition publiquement à l'adresse suivante : `https://github.com/elmontaser1998/TextMine_2026`.

## 4.3 Résultats Principaux

La performance comparative des équipes participantes est résumée dans le Tableau 1. Notre méthode proposée, soumise sous le nom d'équipe **Cursive**, a obtenu la première position sur les classements public et privé.




(a) Nombre d'expansions candidates par exemple d'entraînement.

(b) Nombre d'expansions candidates correctes.

FIG. 3 – *Distributions des expansions candidates et correctes dans l'ensemble d'entraînement.*

| Rang | Équipe | F1 Public | F1 Privé |
|---|---|---|---|
| **1** | **Cursive (Nous)** | **0.9017** | **0.9069** |
| 2 | Mokipo_ | 0.8806 | 0.8814 |
| 3 | AR | 0.8663 | 0.8632 |
| 4 | EDF R&D | 0.8394 | 0.8604 |
| 5 | David | 0.7916 | 0.8419 |
| 6 | Mehdinho | 0.7846 | 0.8108 |
| 7 | yqnis | 0.7768 | 0.7867 |
| 8 | maksimko | 0.8129 | 0.7812 |

TAB. 1 – *Classement final sur les ensembles de test Public et Privé de TextMine'26. Notre soumission **Cursive** (utilisant DACE) surpasse le second d'une marge significative.*

Nous avons obtenu un score F1 Privé final de **0,9069**, surpassant la deuxième meilleure équipe d'environ 2,5 points de pourcentage. Une observation critique est la stabilité de notre modèle : alors que de nombreux concurrents ont vu leurs performances fluctuer entre les partitions Publique et Privée, notre score est resté constant (et s'est même légèrement amélioré sur l'ensemble Privé). Cela indique que le cadre **DACE** empêche le surapprentissage sur des exemples d'entraînement spécifiques et généralise efficacement aux données inédites.

## 4.4 Étude d'Ablation

Afin de mesurer la contribution de chaque composant du pipeline, nous avons mené une étude d'ablation (Tableau 2). Nous comparons des approches à prompt unique au pipeline DACE complet afin d'isoler l'impact du prompting dynamique, de l'architecture du modèle et de l'agrégation en ensemble.

| Méthode | F1 Public | F1 Privé |
|---|---|---|
| Claude 3.5 Sonnet (Template A seul) | 0.8407 | 0.8571 |
| Claude 3.5 Sonnet (Template B seul) | 0.8292 | 0.8368 |
| Claude 3.5 Sonnet + Prompting Dynamique | 0.8876 | 0.8940 |
| Claude 4.1 Opus + Prompting Dynamique | 0.8778 | 0.8852 |
| **DACE (Ensemble Complet)** | **0.9017** | **0.9069** |

TAB. 2 – *Étude d'ablation montrant l'apport incrémental du prompting dynamique et de l'agrégation en ensemble. Les modèles de base utilisent des prompts statiques zero-shot ou few-shot.*

Les résultats indiquent que les stratégies de prompting statiques sont limitées prises isolément. Le baseline few-shot (« Template A seul ») atteint un F1 Privé de 0,8571, mais échoue fréquemment sur les 95 acronymes inédits du test, où l'absence d'exemples d'entraînement induit un biais de fréquence. À l'inverse, la stratégie de correspondance de définition (« Template B seul », F1 Privé 0,8368) réduit les hallucinations pour les acronymes fortement ambigus, mais manque de flexibilité lorsque le contexte est informatif, ce qui pénalise les cas standards.

L'introduction du prompting dynamique produit le gain principal, portant le F1 Privé à 0,8940 grâce à une sélection adaptative des gabarits selon la visibilité de l'acronyme et l'ambiguïté des candidats. Les cas inédits ou fortement ambigus sont dirigés vers le gabarit strict, tandis que les instances usuelles conservent un contexte few-shot, réduisant l'écart de généralisation pour les acronymes à faibles ressources. L'ensemble DACE améliore encore les performances (F1 Privé 0,9069) en agrégeant Claude 3.5 Sonnet et Claude 4.1 Opus, ce qui atténue la variance et les erreurs spécifiques à chaque modèle. La proximité entre les scores Public (0,9017) et Privé (0,9069) témoigne d'une désambiguïsation robuste plutôt que d'un surapprentissage.

# 5 Conclusion

Nous avons présenté DACE, un cadre modulaire combinant prompting dynamique, génération augmentée par récupération, sélection contextuelle et agrégation d'ensemble pour la désambiguïsation d'acronymes dans la documentation ferroviaire spécialisée. Sa conception repose sur deux constats issus du corpus : la présence d'acronymes rares ou inédits exige une généralisation au-delà des exemples supervisés, et la variabilité des candidats et la spécificité sémantique nécessitent un ancrage explicite dans des connaissances techniques fiables. Cette observation a motivé une méthodologie hybride intégrant le raisonnement des LLM à une récupération structurée et un prompting few-shot contrôlé.

Les résultats expérimentaux confirment l'efficacité de DACE, qui obtient le meilleur F1 de la compétition TextMine'26 et des performances stables sur les évaluations publiques et privées. Les ablations montrent que chaque composant contribue significativement : le prompting dynamique réduit les erreurs sur les acronymes ambigus, la récupération renforce l'alignement factuel avec la terminologie ferroviaire, et l'ensembling diminue la variance stochastique.

Au-delà de la compétition, DACE peut s'appliquer à d'autres domaines techniques à terminologie dense et faiblement annotée. Le framework est indépendant du modèle, du domaine et de l'inventaire d'acronymes, facilitant le transfert. Les perspectives incluent l'extension à des scénarios multilingues, l'intégration de graphes de connaissances comme sources de récupération, l'évaluation sur des benchmarks ouverts (e.g., GLADIS), et l'optimisation automatique des prompts ou la sélection d'exemples pilotée par renforcement. Ces résultats illustrent la valeur pratique de combiner prompting adaptatif et techniques de recherche d'information raisonnées pour la désambiguïsation spécialisée.

# References

Chen, L., G. Varoquaux, and F. Suchanek (2023). Gladis: A general and large acronym disambiguation benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 2073–2088.

Kandpal, N., H. Deng, A. Roberts, E. Wallace, and C. Raffel (2023). Large language models struggle to learn long-tail knowledge. In *International conference on machine learning*, pp. 15696–15707. PMLR.

Kong, F. and S. Ahn (2024). Use of knowledge graphs for construction safety management: A systematic literature review. *Information 15*(7), 390.

Kugic, A., S. Schulz, and M. Kreuzthaler (2024). Disambiguation of acronyms in clinical narratives with large language models. *Journal of the American Medical Informatics Association 31*(9), 2040–2046.

Kugic, A., S. Schulz, and M. Kreuzthaler (2025). Embedding-based acronym disambiguation supported by large language models in german clinical narratives.

Lewis, P., E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, et al. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems 33*, 9459–9474.

Li, C., L. Ji, and J. Yan (2015). Acronym disambiguation using word embedding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Volume 29.

Liu, J., D. Shen, Y. Zhang, W. B. Dolan, L. Carin, and W. Chen (2022). What makes good in-context examples for gpt-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd workshop on knowledge extraction and integration for deep learning architectures*, pp. 100–114.

Luce Lefeuvre, Coralie Reutenauer, A. G. P. C. C. L. (2025). Défi textmine / egc 2026. `https://kaggle.com/competitions/defi-text-mine-egc-2026`. Kaggle.

Min, S., X. Lyu, A. Holtzman, M. Artetxe, M. Lewis, H. Hajishirzi, and L. Zettlemoyer (2022). Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*.

Okazaki, N. and S. Ananiadou (2006). Building an abbreviation dictionary using a term recognition approach. *Bioinformatics 22*(24), 3089–3095.

Pan, C., B. Song, S. Wang, and Z. Luo (2021). Bert-based acronym disambiguation with multiple training strategies. *arXiv preprint arXiv:2103.00488*.

Robertson, S., H. Zaragoza, et al. (2009). The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval 3*(4), 333–389.

Sainz, O., O. L. de Lacalle, E. Agirre, and G. Rigau (2023). What do language models know about word senses. *Zero-Shot WSD with Language Models and Domain Inventories*.

Schwartz, A. S. and M. A. Hearst (2002). A simple algorithm for identifying abbreviation definitions in biomedical text. In *Biocomputing 2003*, pp. 451–462. World Scientific.

Shuster, K., S. Poff, M. Chen, D. Kiela, and J. Weston (2021). Retrieval augmentation reduces hallucination in conversation. *arXiv preprint arXiv:2104.07567*.

Song, G., H. Lee, and K. Shim (2022). T5 encoder based acronym disambiguation with weak supervision. *SDU@ AAAI 22*.

Sumanathilaka, D., N. Micallef, and J. Hough (2025). Prompt balance matters: Understanding how imbalanced few-shot learning affects multilingual sense disambiguation in llms. *arXiv preprint arXiv:2510.03762*.

Touvron, H., L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al. (2023). Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Veyseh, A. P. B., F. Dernoncourt, W. Chang, and T. H. Nguyen (2021). Maddog: A web-based system for acronym identification and disambiguation. *arXiv preprint arXiv:2101.09893*.

Veyseh, A. P. B., F. Dernoncourt, T. H. Nguyen, W. Chang, and L. A. Celi (2020). Acronym identification and disambiguation shared tasks for scientific document understanding. *arXiv preprint arXiv:2012.11760*.

Wu, Y., J. Xu, Y. Zhang, and H. Xu (2015). Clinical abbreviation disambiguation using neural word embeddings. In *Proceedings of BioNLP 15*, pp. 171–176.

Zahariev, M. (2004). A (acronyms).

# Résumé

Acronym Disambiguation (AD) is a fundamental challenge in technical text processing, particularly in specialized sectors where high ambiguity complicates automated analysis. This paper addresses AD within the context of the TextMine'26 competition on French railway documentation. We present DACE (**D**ynamic Prompting, Retrieval **A**ugmented Generation, **C**ontextual Selection, and **E**nsemble Aggregation), a framework that enhances Large Language Models through adaptive in-context learning and external domain knowledge injection. By dynamically tailoring prompts to acronym ambiguity and aggregating ensemble predictions, DACE mitigates hallucination and effectively handles low-resource scenarios. Our approach secured the top rank in the competition with an F1 score of 0.9069.