

Design d'invites et aides-correcteurs

AI Horizons in Language Learning - Atelier 2

Pascal Brissette

2026-01-22

Plan de l'atelier (90 min)

- Introduction et contexte (5 min)
 - Partie A : créer un aide-correcteur (60 min)
 - Partie B : des aides aux agents (25 min)
-

Objectif de l'atelier

Transformer un LLM généraliste en aide-correcteur spécialisé

- Comprendre le fonctionnement des modèles de langage
 - Maîtriser le design d'invites (message système et interaction)
 - Découvrir les agents et leurs capacités
-

Partie A

Créer un aide-correcteur

Comment fonctionne un LLM ?

Un système de prédition probabiliste

- Entraîné sur des milliards de textes
 - Prédit le mot suivant le plus probable
 - **N'est pas** un expert qui « comprend » la grammaire
-

Forces et limites pour la correction

Forces

- Repère les types d'erreurs
- Propose des solutions de remplacement
- S'adapte aux registres

Limites

- Peut rater des erreurs
- Peut halluciner des références
- Sensible à la qualité des instructions

Message clé : la qualité du résultat dépend du modèle **ET** des instructions

Anatomie d'une interaction

Deux niveaux d'instructions

Message système

- Instruction **permanente** qui « colle » à la conversation
- Définit les rôles, règles et contraintes
- Invisible pour l'utilisateur

Invite (prompt)

- Requête **spécifique**
 - Varie selon la tâche
 - Visible dans l'interface
 - Évolue au cours de l'interaction
-

Le cadre RTF

Structure d'un message système efficace

1. Rôle : définir la *persona* du LLM, son expertise, ainsi que l'utilisateur type
 2. Tâche : préciser la tâche, les règles particulières et contraintes
 3. Format : format attendu de la réponse (tableau, liste, markdown, texte brut, etc.)
-

R - Rôle

Définir l'identité et l'expertise

- Quelle est la *persona* du LLM?
- Quelle est sa spécialité?
- Quelles qualités particulières possède-t-il?

Tu es un aide-correcteur de français langue seconde. Tu possèdes une expertise en...

T - Tâche

Préciser la tâche et les contraintes

- Que doit faire l'assistant ?
 - Quelles règles doit-il respecter ?
 - Qu'est-ce qu'il ne doit **pas** faire ?
-

F - Format

Structurer la réponse attendue

- Comment organiser la réponse ?
- Quel niveau de détail ?



Tip

Pour un aide-correcteur, un format de sortie de type «tableau» est efficace.

Outil : Prompt Canvas

The screenshot shows the 'Prompt Canvas' application interface. At the top, it says 'Prompt Canvas' and 'Prompt engineering with reusable templates in YAML format.' Below that are three buttons: 'Prompt Generator' (highlighted in blue), 'Schema Editor', and 'Guide'. A 'Logout' icon is in the top right corner. The main area has a dark background with white text. It starts with a 'Select a Template' section where 'R-T-F Framework' is selected from a dropdown, and there are 'Upload' and 'Download with Values' buttons. Below this is a 'Role' section with the sub-instruction: 'Define the persona or role the AI should adopt (e.g., "You are a travel blogger").' A text input field contains the French sentence: 'Tu es un correcteur linguistique de français qui lit avec attention et corrige des travaux universitaires écrits par des étudiants.' Underneath is a 'Task' section with the sub-instruction: 'Describe the main task the AI needs to accomplish in detail.' A text input field contains the French sentence: 'Tu mobilises ta connaissance approfondie de la langue française pour corriger les erreurs de langue et d'usage dans le texte qui t'est soumis.'

Aide à la construction de messages système

promptcanvas.ml4den.com

Activité pratique

Créer votre aide-correcteur

Questions

- Le correcteur a-t-il une expertise particulière?
 - Quel type de texte doit-il corriger?
 - Doit-il simplement proposer une correction?
 - Doit-il fournir des références?
-

Phase 1 : Rédaction

Consignes

1. Établir, pour l'exercice, un contexte d'enseignement commun.
 2. Rédiger un message système complet (RTF).
 3. Utiliser Prompt Canvas si cela est utile.
-

Phase 2 : Partage

Discussion collective

- Partage des messages en groupes de 2-3 et retour.

Phase 3 : vérifier l'efficacité du message

- Si vous avez suivi le premier atelier, vous pouvez utiliser AnythingLLM et un LLM ouvert.
- Dans le cas contraire, utilisez Copilot (McGill) ou LeChat (Mistral)

AnythingLLM

The screenshot shows the AnythingLLM application interface. At the top, there's a toolbar with standard file operations like File, Edit, View, Window, Help, and a version indicator (v1.5). Below the toolbar is a "QUICK LINKS" section with three buttons: "Send Chat", "Embed a Document", and "+ Create Workspace". Underneath is a "EXPLORE MORE FEATURES" section with three cards: "Custom AI Agents" (Chat using @agent, Build an agent flow), "Slash Commands" (Create a Slash Command, Explore on Hub), and "System Prompts" (Modify a System Prompt, Manage prompt variables). Below these are "UPDATES & ANNOUNCEMENTS" cards for "Private Web Scraping", "AnythingLLM Mobile Beta", and "Community Hub updates". At the bottom left is a "RESOURCES" section with links to Docs, Star on Github, and Keyboard Shortcuts. The main area is mostly empty, suggesting a clean slate or a specific feature page.

CoPilot

The screenshot shows the Microsoft CoPilot interface. At the top, it displays the URL <https://m365.cloud.microsoft/chat?auth=2&home=1>. The main area features a "New chat" button and a sidebar with options like Rechercher, Bibliothèque, Créer, and Enseigner. A message from "Copilot Chat" says: "Voici le texte à corriger: ""Aujourd'...". Below this is a search bar and a "Clavardages" section with a "Tous les clavardages" link. The central part of the screen shows a message from "Copilot" asking "Bonjour! Comment puis-je vous aider?". Below this is a text input field "Écrivez un message à Copilot" with a "+" button. To the right are three task cards: "List key points from Proposition courriel au recteur.docx", "Draft an email to my team asking for feedback on Document de...", and "Summarize this: upload file". A "Voir plus" link is located at the bottom right of these cards. At the bottom left, there's a sidebar with "Applications" and a profile for "Pascal Brissette". A small pop-up window in the bottom right corner says "Nous aimerions avoir votre avis. Accepteriez-vous de répondre à deux questions ?".

- Utiliser l'un des deux textes «à corriger» du répertoire GitHub

- Quels sont les aspects intéressants du résultat?
 - Que pourrait-on améliorer ?
 - Analyser un message système fonctionnel
-

Phase 4 : révision

Améliorer votre message

- Intégrez les suggestions de la discussion
 - Affinez votre message système
 - Insérez-le dans votre interface
 - LeChat : Instructions → Personnaliser
 - AnythingLLM : → Paramètres du chat → Invite
-

Mise à l'épreuve: testez la V2 du message système

Évaluer la qualité

- **Précision** : les erreurs sont-elles bien identifiées ?
 - **Pertinence** : les corrections sont-elles appropriées ?
 - **Clarté** : les explications sont-elles compréhensibles ?
 - **Utilité pédagogique** : contribue-t-il vraiment à l'apprentissage ?
 - **Problèmes** : hallucinations ? Ton inadapté ?
-

Partie B

Des aides aux agents

Le problème des références

Démonstration

Demander à un LLM : > “Corrige l'erreur suivante et fournis la règle grammaticale...”

Résultat possible : références inventées, numéros de pages fictifs, sources inexistantes

Problème : crédibilité compromise en contexte pédagogique

Aide vs Agent

Aide

- LLM seul
- Message système
- Répond directement
- Peut inventer des sources

Agent RAG

- LLM + outils
- Message système
- Recherche dans une base
- Cite des sources réelles

Différence clé : l'agent peut consulter des documents avant de répondre

Composantes d'un agent

Architecture

1. **LLM** : modèle de langage (Mistral, etc.)
2. **Message système** : instructions de correction
3. **Base documentaire** : 2270 fiches OQLF
4. **Système RAG** : recherche et citation des sources

Démonstration : agent correcteur

Avec base OQLF

- 2270 fiches de la Banque de dépannage linguistique
- Recherche sémantique automatique
- Références exactes



Note

Démonstration

Vérification

Les références sont réelles

- Possibilité d'ouvrir la fiche citée
- Confirmation du contenu



Tip

Utile pour justifier une correction auprès des étudiants

Avantages et limites

Avantages

- Crédibilité accrue
- Sources vérifiables
- Références officielles
- Ajout de sources

Limites

- Plus complexe à mettre en place
- Dépend de la base documentaire
- Pas toujours nécessaire
- Temps requis (latence) (plusieurs appels LLM)

Bonnes pratiques

À retenir

- Tester plusieurs fois votre message système
 - Rester critique à l'égard des corrections proposées
 - Vérifier les références grammaticales (surtout sans agent)
 - Utilisez le modèle le plus simple qui fonctionne
-

Ressources disponibles

Sur GitHub

- [Texte de test pour la correction](#)
- [Exemples de messages système](#)
- [Présentation du premier atelier, avec instructions pour installer Ollama/AnythingLLM](#)

i Note

github.com/pbriss7/Atelier-Design-Invites-Agents

Pour aller plus loin

Expérimentations possibles

- Différentes températures (revoir atelier 1)
 - Plusieurs assistants spécialisés (correction, rétroaction, évaluation)
 - Autres bases documentaires (Usito, ressources départementales, etc.)
 - Agents plus complexes (multi-outils)
-

Questions et discussion



Contact

pascal.brissette@mcgill.ca