LARCHER Elya, MARROT Mathilde, MONIN Louis,

VALIGNON Martin, WARLET Antoine

ING2 MI2 – Mathématiques Appliquées

Classification d'expériences professionnelles LinkedIn par les techniques « Few-shot learning »

Table des matières

Contextualisation

Le projet de recherches mené par le CNRS et plusieurs universités, dont CY Cergy Paris Université vise à créer une base de données européenne des travailleurs dans le domaine de l'intelligence artificielle. Ce projet a pour objectif d'utiliser LinkedIn comme source de données. On considère un échantillon initial de 2500 profils avec une expérience en IA constitué via un bot de scraping. Néanmoins, cette méthode de classification des expériences professionnelles s'avère difficile notamment en raison de la présence d'expériences non pertinentes.

On peut considérer d'autres méthodes de classification basées sur les mots-clés, le machine learning basique et le deep learning avec des architectures Transformers comme Bert ont été testées. Ces méthodes présentent de bons résultats mais elles requièrent un grand nombre de données d'exemple pour l'entraînement, ce qui implique un effort de labellisation important.

Pour ainsi envisager de surmonter ces difficultés, le projet envisage d'explorer les techniques de « few-shot learning », qui pourraient fournir des résultats satisfaisants avec moins d'effort de labellisation. L'objectif est de réduire les ressources nécessaires pour la labellisation tout en maintenant l'efficacité de la classification des expériences professionnelles sur LinkedIn.

Notre base de données est un fichier contenant environ 11 000 expériences professionnelles LinkedIn provenant d'environ 2 000 profils, chaque profil contient 9 variables en colonnes qui sont l'id, companyName, companyUrl, jobTitle, dataRange, location, description, logoUrl et label.

Objectifs

Revue de la littérature des techniques few-shot learning

Définition

Le few-shot learning (FSL) est un problème d'apprentissage où la base de données contient seulement quelques exemples labélisés de chaque classe. Cette technique est surtout utilisée en vision par ordinateur (domaine scientifique qui est une branche de l'intelligence artificielle qui traite de la manière dont les ordinateurs peuvent acquérir une compréhension de haut niveau à partir d'images ou de vidéos numériques. Du point de vue de l'ingénierie, il cherche à comprendre et à automatiser les tâches que le système visuel humain peut effectuer) [1] car elle présente des approches et des résultats utilisables dans ce domaine. Apprendre avec peu de données peut être abordé soit par l'intermédiaire des données elles-mêmes, soit par l'utilisation d'algorithmes dédiés, soit grâce à l'adaptation directe d'un modèle. [2]

Méthodes

Transfer Learning avec un modèle de Langue pré-entraîné

C'est une méthode puissante et adaptable, particulièrement adaptée pour traiter des données textuelles comme celles de notre base de données LinkedIn.

Fonctionnement de la méthode

Le Pré-entraînement : Les modèles comme BERT, GPT ou RoBERTa ont été préalablement entraînés sur d'énormes corpus de texte. Cela leur permet d'avoir une compréhension avancée de la langue, de la structure grammaticale, du contexte et des nuances sémantiques.

Fine-Tuning : Cette étape consiste à ajuster le modèle pré-entraîné sur notre ensemble de données spécifiques. On commence par transformer les descriptions de postes et d'expériences professionnelles en vecteurs de caractéristiques compréhensibles par le modèle. Ensuite, le modèle est légèrement modifié et entraîné sur ces vecteurs avec les labels correspondant de 0 à 4.

Prédiction : Après l'entraînement, le modèle est capable de prendre de nouvelles descriptions d'expériences professionnelles et de prédire leur label. Il détermine si une expérience est liée à l'IA (label 4) ou non (label 0), avec divers degrés d'incertitude pour les labels intermédiaires.

Étapes du processus

1. **Préparation des Données** :
   - **Nettoyage** : Les descriptions d'expériences professionnelles sont nettoyées pour enlever les caractères non pertinents, corriger les fautes, et normaliser le texte.
   - **Tokenisation** : Le texte est divisé en tokens (mots ou sous-unités de mots). Ceci est crucial pour que le modèle puisse traiter le texte.

2. **Transformation en Vecteurs** :
   - Chaque token est transformé en un vecteur numérique à l'aide du modèle pré-entraîné. Ces vecteurs représentent les caractéristiques sémantiques et syntaxiques des mots dans un espace à haute dimension.

      1. **Tokenisation :**
         - La première étape consiste à diviser chaque description (par exemple, la description du poste) en unités plus petites appelées "tokens". Les tokens peuvent être des mots, des sous-unités de mots (n-grams), ou même des caractères, en fonction de la granularité souhaitée.
         - Exemple : La phrase "Développeur en intelligence artificielle" pourrait être tokenisée en ["Développeur", "en", "intelligence", "artificielle"].

      2. **Embeddings de Mots :**
         - Chaque token est ensuite converti en un vecteur numérique, généralement

appelé "embedding de mot". Ces embeddings sont extraits à partir du modèle de langue pré-entraîné.
- Les embeddings de mots capturent les caractéristiques sémantiques et syntaxiques des mots en se basant sur le contexte dans lequel ils apparaissent dans le vaste corpus de texte sur lequel le modèle a été pré-entraîné.
- Les embeddings permettent au modèle de représenter chaque mot sous forme de vecteur numérique dans un espace multidimensionnel.

3. **Vecteurs de Phrases ou de Documents :**
    - Pour obtenir une représentation globale de la description du poste, les embeddings de chaque token dans la phrase sont souvent combinés d'une manière spécifique. Par exemple, on peut prendre la moyenne des embeddings ou les concaténer.
    - Cela donne lieu à un vecteur numérique représentant l'ensemble de la description. Ce vecteur contient des informations sur la sémantique globale et la structure syntaxique du texte.

4. **Contextualisation (si nécessaire) :**
    - Certains modèles de langage pré-entraînés, tels que BERT et GPT, prennent en compte le contexte des mots environnants. Cela signifie que l'embedding d'un mot peut varier en fonction de son contexte dans la phrase. Cette contextualisation permet de capturer des nuances et des sens spécifiques en fonction du contexte.
    - Pour chaque mot, le modèle ajuste l'embedding en fonction du contexte global de la phrase.

5. **Matrice de Vecteurs :**
    - L'ensemble des descriptions transformées est représenté sous forme d'une matrice, où chaque ligne correspond à un individu et chaque colonne représente une dimension du vecteur.
    - Cette matrice est utilisée comme entrée pour le fine-tuning du modèle.

3. **Contextualisation** :
    - Le modèle utilise le contexte de chaque mot (les mots environnants) pour ajuster les vecteurs. Cela permet de capturer le sens spécifique des mots dans différents contextes. Des caractéristiques (features) sont extraites à partir des données pour aider l'algorithme à différencier les profils pertinents des non pertinents. Cela peut inclure l'analyse de la sémantique des titres de poste, la durée des expériences, la localisation, et le contenu des descriptions de poste.

4. **Extraction de Caractéristiques** :
    - Le modèle génère une représentation vectorielle globale pour chaque description d'expérience professionnelle. Cette représentation est une synthèse des caractéristiques clés du texte.

5. **Fine-Tuning** :
    - Le modèle est ensuite affiné avec notre jeu de données. Cela implique l'entraînement du modèle sur les vecteurs de texte avec les labels correspondants. L'objectif est

d'ajuster les poids du modèle pour qu'il puisse prédire correctement les labels de vos données.

6. **Classification** :
    - Une fois le modèle affiné, il peut classer de nouvelles descriptions d'expérience en prédisant leur label (de 0 à 4). Le modèle évalue la probabilité qu'une expérience appartienne à chaque catégorie de label.

Pourquoi la méthode devrait fonctionner ?

Au niveau de la compréhension contextuelle, les modèles de langage naturel comprennent le contexte et les nuances de langage, ce qui est essentiel pour analyser les descriptions d'expériences professionnelles qui peuvent varier en style, en type de langage et en détails.

Par rapport à la capacité d'adaptation, le fine-tuning permet d'adapter le modèle aux spécificités de notre base de données. Même si les descriptions de poste ne sont pas standardisées, le modèle peut apprendre à identifier les caractéristiques pertinentes pour la classification.

En termes d'efficacité, les modèles pré-entraînés ont déjà une connaissance approfondie de la langue, ils requièrent donc moins de données spécifiques pour l'entraînement. Cela les rend idéaux pour des bases de données de taille modeste, comme notre base de données.

Enfin, ces modèles sont très flexibles, ils peuvent gérer différents formats de texte et s'adapter à divers domaines, ce qui est crucial pour une base de données variée comme celle des expériences professionnelles.

ProtoNets (Prototypical Networks)

Fonctionnement de la méthode

1. **Extraction de Caractéristiques** :
    - **Encodage** : Utilisez un modèle de traitement du langage naturel (NLP) pour transformer les descriptions textuelles des expériences professionnelles en vecteurs de caractéristiques. Ces vecteurs capturent le contenu sémantique et contextuel des descriptions.

2. **Création de Prototypes** :
    - **Groupement** : Pour chaque catégorie (par exemple, IA et non-IA), les vecteurs des exemples dans la catégorie sont moyennés pour créer un "prototype" représentatif de cette catégorie.
    - **Représentation** : Chaque prototype est une représentation vectorielle moyenne des exemples de sa catégorie, capturant les traits saillants communs à ces exemples.

3. **Classification** :
    - **Calcul de Distance** : Pour un nouvel exemple, le modèle calcule la distance (par

exemple, distance euclidienne) entre le vecteur de cet exemple et chaque prototype.
- **Attribution de Catégorie** : L'exemple est classé dans la catégorie dont le prototype est le plus proche en termes de distance.

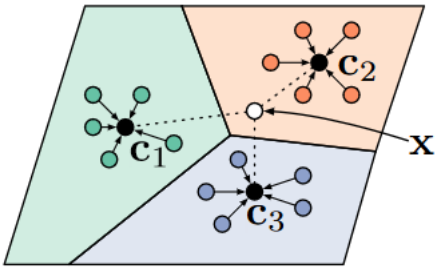
Étapes du processus

1. **Préparation des Données** :
   - **Nettoyage** : Les descriptions des expériences professionnelles sont nettoyées pour enlever les caractères inutiles, les erreurs, et uniformiser le texte.
   - **Tokenisation et Encodage** : Le texte est divisé en tokens (mots ou sous-unités de mots), qui sont ensuite convertis en vecteurs numériques à l'aide d'un modèle de NLP.

2. **Extraction de Caractéristiques** :
   - **Création de Vecteurs** : Chaque description d'expérience professionnelle est transformée en un vecteur de caractéristiques, qui capture les aspects sémantiques et contextuels de la description.

3. **Formation des Prototypes** :
   - **Regroupement** : Pour chaque catégorie (IA, probablement IA, incertain, etc.), les vecteurs de cette catégorie sont moyennés pour former un prototype unique. Ce prototype représente le "centre" de la catégorie dans l'espace des caractéristiques.

4. **Processus de Classification** :
   - **Calcul de Distance** : Pour chaque nouvelle expérience, le modèle calcule la distance entre son vecteur de caractéristiques et chaque prototype.
   - **Attribution de Classe** : L'expérience est classée dans la catégorie dont le prototype est le plus proche, en termes de distance.



[3]

Exemple

Pour illustrer la méthode des réseaux prototypiques (ProtoNets) à l'aide d'un exemple de la base de données, prenons l'expérience professionnelle suivante extraite de notre fichier CSV :

- **ID**: 551303934_0
- **Entreprise**: ALTEN
- **Intitulé du poste**: Stagiaire ingénieur en intelligence artificielle
- **Période**: juil. 2021 – Aujourd'hui

- **Lieu**: Rennes, Bretagne, France
- **Description**: Conception et programmation pour une plateforme d'intelligence artificielle.

Suivons les étapes clés de la méthode ProtoNets pour classer cette expérience :

1. Préparation des Données

La description de l'expérience est nettoyée et tokenisée. Imaginons qu'après nettoyage, nous avons les tokens significatifs tels que "conception", "programmation", "plateforme", et "intelligence artificielle".

2. Extraction de Caractéristiques

Ces tokens sont ensuite encodés en vecteurs numériques à l'aide d'un modèle de NLP. Supposons que le vecteur résultant pour cette description soit `[0.45, -0.22, 0.88, 0.10]`, qui capture les aspects sémantiques et contextuels de la description.

3. Création de Prototypes

Imaginons que nous ayons déjà des prototypes pour deux catégories : IA (Intelligence Artificielle) et non-IA. Le prototype IA pourrait être le vecteur moyen `[0.40, -0.20, 0.85, 0.15]` et le prototype non-IA `[0.10, 0.30, -0.50, 0.05]`.

4. Classification

La distance entre le vecteur de l'expérience `[0.45, -0.22, 0.88, 0.10]` et chaque prototype est calculée. Supposons que la distance euclidienne jusqu'au prototype IA soit plus petite que celle jusqu'au prototype non-IA, ce qui signifie que l'expérience est plus proche du prototype IA.

Ainsi, cette expérience serait classée dans la catégorie IA en se basant sur sa proximité avec le prototype de cette catégorie.

Optimisation du modèle pour nos données

Sélection de Caractéristiques : Identifiez les aspects les plus informatifs des descriptions d'expériences pour améliorer la création des vecteurs de caractéristiques.

Choix de Métrique de Distance : Expérimentez avec différentes métriques de distance (Euclidienne, Manhattan, ...) pour trouver celle qui offre les meilleures performances de classification.

Validation Croisée : Utilisez la validation croisée pour évaluer la robustesse du modèle et son aptitude à généraliser sur des données non vues.

Analyse des Erreurs : Examinez les erreurs de classification pour comprendre les limites du modèle et ajuster les étapes de prétraitement ou le calcul des prototypes.

Pourquoi la méthode devrait fonctionner ?

Au niveau de la représentation concentrée, en créant des prototypes, les ProtoNets concentrent les informations essentielles de chaque catégorie. Cela aide le modèle à identifier les caractéristiques clés qui définissent chaque catégorie.

L'efficacité dans la gestion des petits ensembles de données, les ProtoNets sont conçus pour le few-shot learning, ce qui signifie qu'ils sont efficaces même avec un petit nombre d'exemples d'entraînement.

Les prototypes moyens permettent au modèle de mieux gérer la variabilité et les subtilités dans les descriptions d'expériences professionnelles.

Enfin, au niveau de la capacité de généralisation, les ProtoNets sont bons pour généraliser à partir de peu d'exemples, ce qui est crucial pour notre base de données où chaque expérience professionnelle est unique.

Siamese Networks

Fonctionnement de la méthode

Au niveau de l'architecture des réseaux, on parle de réseaux siamois (identiques) car les réseaux siamois se composent de deux réseaux neuronaux identiques, chacun prenant une entrée différente. Ces réseaux partagent les mêmes poids et structure.

Le traitement de l'information se fait de manière parallèle, chaque réseau traite son entrée, dans notre cas une description d'expérience professionnelle, indépendamment produisant un vecteur de caractéristiques pour chaque description.

Par la suite, nous avons l'extraction de caractéristiques, avec l'encodage de texte, les descriptions sont converties en vecteurs de caractéristiques à l'aide d'un modèle de NLP, capturant l'essence sémantique et contextuelle des textes.

Enfin, vient la comparaison et l'évaluation de la similarité, avec le calcul de distance, la similarité entre les vecteurs de caractéristiques issus des deux réseaux est évaluée, souvent via une distance euclidienne ou une autre métrique de similarité. Puis le modèle évalue si les deux expériences sont similaires ou non, basé sur la proximité des vecteurs de caractéristiques.

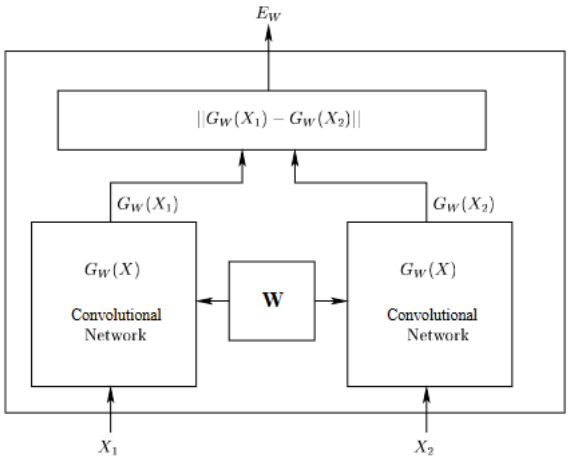
Étapes du processus

1. **Préparation des Données** :
    - **Nettoyage et Prétraitement** : Les descriptions d'expériences professionnelles sont d'abord nettoyées et pré traitées pour enlever les éléments non pertinents et uniformiser le texte.
    - **Sélection de Paires** : Formez des paires d'expériences professionnelles. Ces paires peuvent être des expériences similaires (même catégorie) ou différentes (catégories variées).

2. **Encodage des Données** :
    - **Transformation en Vecteurs** : Utilisez un modèle de NLP pour convertir les descriptions textuelles en vecteurs de caractéristiques. Ces vecteurs représentent les aspects sémantiques et contextuels des descriptions.

3. **Traitement par les Réseaux Siamois** :
    - **Passage à travers les Réseaux** : Chaque description dans une paire est traitée par un des deux réseaux siamois. Ces réseaux partagent la même architecture et les mêmes poids.
    - **Extraction de Caractéristiques** : Chaque réseau produit un vecteur de caractéristiques pour sa description respective.

4. **Comparaison des Vecteurs** :
    - **Calcul de Distance ou de Similarité** : Le modèle calcule une mesure de distance ou de similarité entre les vecteurs de caractéristiques issus des deux réseaux.
    - **Évaluation de la Similarité** : Basé sur cette mesure, le modèle détermine si les deux expériences sont similaires ou non.



[4]

Exemple

Étape 1 : Préparation des Données

Les descriptions d'expériences professionnelles sont nettoyées pour éliminer les éléments non pertinents et uniformiser le texte. Par exemple, les stop-words (mots très courants qui n'apportent pas de valeur significative) sont retirés, et les mots sont ramenés à leur forme de base (lemmatisation).

Profil 1 : "Conception et programmation pour une plateforme d'intelligence artificielle."
Profil 2 : "Développement et mises à jour d’applications commerciales."

Étape 2 : Encodage des Données

Utilisant un modèle de NLP, les descriptions textuelles sont converties en vecteurs de

caractéristiques. Imaginons que nous utilisons un modèle comme BERT ou Word2Vec pour obtenir des représentations vectorielles de chaque description.

Vecteur Profil 1 : [0.65, -0.32, 0.47, ...] (pour simplification, représentation en dimension réduite)
Vecteur Profil 2 : [0.58, -0.27, 0.50, …]

Les vecteurs sont définis par l'encodage de textes à l'aide de modèles de NLP, qui transforment les descriptions textuelles en vecteurs numériques représentant sémantiquement le contenu. Ces vecteurs peuvent être générés par des méthodes comme one-hot encoding, word embeddings (Word2Vec, GloVe), ou contextual embeddings (BERT, GPT), et sont utilisés pour calculer des similarités ou des distances entre les exemples traités par les réseaux.

Étape 3 : Traitement par les Réseaux Siamois

Chaque vecteur de caractéristiques est traité par son propre réseau siamois. Bien que les réseaux soient identiques en structure et partagent les mêmes poids, ils traitent les vecteurs indépendamment, affinant les caractéristiques pour mettre en évidence les aspects uniques de chaque expérience professionnelle.

Étape 4 : Comparaison des Vecteurs

Après le passage à travers les réseaux, nous obtenons des vecteurs de caractéristiques finaux pour chaque profil. Supposons maintenant que le modèle calcule la distance euclidienne entre ces vecteurs pour évaluer leur similarité.

Calcul de Distance : Supposons que la distance calculée soit relativement faible, indiquant que les vecteurs sont assez proches dans l'espace vectoriel.

Étape 5 : Évaluation de la Similarité

Basé sur la distance calculée, si nous avions défini un seuil de similarité à l'avance (par exemple, une distance inférieure à un certain valeur indique une similarité), le modèle pourrait conclure que les deux expériences sont similaires car elles impliquent toutes deux le développement de logiciels, bien qu'elles soient appliquées dans des domaines légèrement différents (intelligence artificielle vs applications commerciales).

Pourquoi la méthode devrait fonctionner ?

Les réseaux siamois sont spécifiquement conçus pour les tâches de comparaison, ce qui les rend idéaux pour évaluer la similarité entre des descriptions d'expériences.

Cette approche peut gérer efficacement la diversité et les variations subtiles dans les descriptions, ce qui semble nécessaire pour notre base de données.

Les réseaux siamois peuvent être adaptés pour fonctionner avec différents types de données textuelles et peuvent être affinés pour répondre aux spécificités de notre projet.

Conclusion

La mise en œuvre de techniques de few-shot learning, telles que le Transfer Learning avec modèles de langue pré-entraînés, les ProtoNets, et les réseaux siamois, présente une voie prometteuse pour surmonter les défis de classification des expériences professionnelles dans le domaine de l'intelligence artificielle sur LinkedIn, avec un effort de labellisation réduit. Ces méthodes offrent des avantages significatifs en termes de compréhension contextuelle, d'adaptabilité, d'efficacité, et de capacité de généralisation, essentiels pour traiter des données textuelles variées et non standardisées.

Bibliographie

[1] https://fr.wikipedia.org/wiki/Vision_par_ordinateur

[2] https://theses.hal.science/tel-03143123v1/document trouvé grâce à https://www.theses.fr/

[3] https://papers.nips.cc/paper_files/paper/2017/file/cb8da6767461f2812ae4290eac7cbc42-Paper.pdf

[4] http://yann.lecun.com/exdb/publis/pdf/chopra-05.pdf