Exercices sur word embeddings

Voici quelques exercices pour renforcer la compréhension des plongements lexicaux (word embeddings) et encourager la pratique des connaissances en programmation et en TALN:

1. Exploration de vecteurs de mots
Se familiariser avec les opérations vectorielles et la similarité sémantique.

Exo : Utiliser un ensemble de vecteurs de mots pré-entraînés (par exemple, Word2Vec ou GloVe) pour trouver les mots les plus proches (similaires) à un mot donné.
Par exemple, demander quel est le mot le plus similaire à "véhicule".

2. Analogies de mots
Comprendre les relations sémantiques 'capturées' par les vecteurs de mots.

Exo : Compléter les analogies en utilisant des vecteurs de mots. Utiliser les opérations vectorielles pour trouver le mot manquant.
Par exemple, "Paris est à France, comme Rome est à ?".

3. Visualisation de l'espace sémantique
Visualiser l'espace vectoriel pour comprendre comment les mots sont organisés sémantiquement.

Exo : Utiliser une méthode de réduction de dimensionnalité comme t-SNE ou PCA pour projeter des vecteurs de mots dans un espace à deux dimensions, puis visualisez-les.
Identifier des clusters de mots et expliquer ce qui les regroupe (par exemple, des mots relatifs à la nourriture, des noms de pays, etc.).

4. Classification de texte
Appliquer des vecteurs de mots dans une tâche de TALN.

Exo : Fournir un ensemble de données de textes courts catégorisés (par exemple, tweets classés par sentimen, ou avis selon polarité +/-).
Représenter ces textes à l'aide de vecteurs de mots et développer un classificateur simple (par exemple, régression logistique) pour prédire la catégorie de textes non vus.

5. Détecter le changement de sens
Observer comment le sens des mots peut varier selon le corpus.

Exo : Donner deux corpus de domaines différents. Entraîner des modèles de plongements lexicaux séparément sur chaque corpus et identifier les mots dont les vecteurs ont le plus changé.
Cela peut montrer comment le sens ou l'utilisation de certains mots peut évoluer.

6. Exploration créative
Encourager l'exploration libre et la découverte de propriétés intéressantes des plongements lexicaux.

Exo : Organiser une "chasse au trésor" où il faut trouver des curiosités et des anomalies dans l'espace sémantique, comme des mots étonnamment proches ou des analogies inattendues.
Cela peut être fait en groupe pour stimuler la discussion et le partage de découvertes.

Réalisation

Pour réaliser ces exercices sur les plongements lexicaux (word embeddings), plusieurs ressources et outils sont nécessaires. Voici une liste des supports recommandés pour faciliter l'apprentissage et la mise en pratique :

Corpus

- Des corpus « généraux » (majoritairement en anglais) tels que Wikipedia, Google News, ou des collections de textes issus de Common Crawl (https://commoncrawl.org/) sont souvent utilisés pour entraîner des modèles de plongements lexicaux généralistes. Pour des tâches spécifiques, choisir un corpus pertinent au domaine d'étude.
- Corpus spécifique au domaine : pour l'exercice de détection de changement de sens ou pour des tâches spécialisées, choisissez un corpus spécifique au domaine (par exemple, articles scientifiques pour le domaine médical, critiques de films pour l'analyse de sentiments).
- Ensembles de données pré-classifiés : pour des exercices de classification de texte, utilisez des ensembles de données avec des étiquettes pré-définies, comme IMDb pour la classification de sentiments ou AG News pour la classification de nouvelles.

Outils et bibliothèques

- Python : Langage de programmation facile à apprendre et riche en bibliothèques pour le TALN et l'apprentissage automatique.
- NLTK / SpaCy: Bibliothèques pour le prétraitement de texte (tokenisation, lemmatisation, suppression des mots vides).
- Gensim: Offre une implémentation efficace de Word2Vec et GloVe, ainsi que des fonctionnalités pour charger des vecteurs de mots pré-entraînés.
- Scikit-learn : Pour la modélisation et l'évaluation de classificateurs dans les exercices de classification de texte.
- TensorFlow / PyTorch : Ces frameworks d'apprentissage profond offrent plus de flexibilité pour des modèles complexes et l'utilisation de réseaux de neurones.
- t-SNE (via Scikit-learn) ou PCA : Outils de réduction de dimension pour la visualisation des plongements lexicaux.

Vecteurs de mots pré-entraînés

- Utiliser des vecteurs de mots pré-entraînés peut grandement simplifier les exercices, surtout pour les groupes avec une expérience limitée en programmation ou en apprentissage automatique. Des modèles pré-entraînés sur de grands corpus sont disponibles pour Word2Vec, GloVe, et FastText. Vous pouvez les télécharger depuis les sites officiels ou les charger directement avec Gensim.

Environnement de développement

- Jupyter Notebook : Offre un environnement interactif pour exécuter du code Python, visualiser des données, et écrire des notes, ce qui est idéal pour l'enseignement et l'apprentissage.
- Google Colab : Environnement basé sur le cloud qui fournit gratuitement des ressources de calcul et facilite le partage de notebooks.

En combinant ces ressources, on a un ensemble complet d'outils pour explorer les plongements lexicaux, comprendre leur fonctionnement et appliquer ces concepts à des tâches réelles de TALN.

Pour démarrer le premier exercice sur l'exploration de vecteurs de mots, par exemple on pourrait suivre les étapes suivantes :

Étape 1 : Choisir les vecteurs de mots

Commencer par décider si l'on veut entraîner ses propres vecteurs de mots ou utiliser des vecteurs pré-entraînés. Pour un premier exercice, et pour simplifier, choisir des vecteurs pré-entraînés. Gensim offre un accès facile à plusieurs ensembles de vecteurs de mots pré-entraînés tels que Word2Vec de Google News, GloVe de Stanford, ou encore FastText.

Étape 2 : Installer les bibliothèques nécessaires

S'assurer que toutes les bibliothèques nécessaires sont installées. Cela inclut `gensim` pour les plongements lexicaux, et `nltk` ou `spacy` pour le traitement de texte si nécessaire. On pourrait les installer via `pip` :

```
pip install gensim nltk spacy
```

Étape 3 : Charger les vecteurs de mots

- Charger les vecteurs de mots pré-entraînés à l'aide de `gensim`. Par exemple, pour charger les vecteurs de GloVe :

```
from gensim.scripts.glove2word2vec import glove2word2vec
from gensim.models.keyedvectors import KeyedVectors
```

- Convertir le format GloVe au format Word2Vec

```
glove_input_file = 'glove.6B.100d.txt'
word2vec_output_file = 'glove.6B.100d.txt.word2vec'
glove2word2vec(glove_input_file, word2vec_output_file)
```

- Charger les vecteurs convertis

```
model = KeyedVectors.load_word2vec_format(word2vec_output_file, binary=False)
```

Étape 4 : Explorer les vecteurs de mots

Maintenant que l'on a chargé les vecteurs, on pourrait commencer à explorer les relations sémantiques entre les mots. Par exemple, on pourrait chercher les mots les plus similaires à un mot donné :

```
print(model.most_similar('france'))
```

Cette commande afficherait les mots les plus proches de "france" dans l'espace vectoriel, ce qui nous donnerait une idée de la manière dont les vecteurs de mots capturent la signification et les relations entre les mots.

Étape 5 : Documenter et réfléchir

Tout en faisant l'exercice, on prendra des notes sur ces découvertes et réflexions. Cela pourrait inclure des surprises sur la manière dont certains mots sont liés, ou des idées sur pourquoi certains

mots ne sont pas aussi proches que l’on aurait attendu. On pourrait utiliser un Jupyter Notebook pour combiner le code, les résultats et les notes en un seul document.

Conseil supplémentaire

Je vous conseille d'expérimenter avec différents mots et d'explorer divers aspects des vecteurs de mots, comme effectuer des opérations vectorielles pour résoudre des analogies de mots. Cela vous donnerait une compréhension plus profonde de la puissance et des limites des plongements lexicaux.