جامعة مولاي إسماعيل
UNIVERSITÉ MOULAY ISMAÏL

المدرسة الوطنية العليا للفنون و المهن
ÉCOLE NATIONALE SUPÉRIEURE D'ARTS ET MÉTIERS

RESUME PARSER

Réalisé par :

BAGUIAN HAROUNA

DIASSANA FATOUMATA

Encadré par :

MR Hajji TARRIK

I. Abstract

Analysez les informations d'un CV à l'aide du traitement du langage naturel, trouvez les mots-clés, regroupez-les dans des secteurs en fonction de leurs mots-clés et enfin montrez le CV le plus pertinent à l'employeur en fonction de la correspondance des mots-clés. Tout d'abord, l'utilisateur télécharge un CV sur la plateforme Web. L'analyseur analyse toutes les informations nécessaires du CV. Une fois que l'utilisateur a confirmé, le CV est enregistré dans notre base de données NoSQL prêt à se montrer aux employeurs.

Mots-clés : analyseur de CV, exploration de texte, traitement du langage naturel, CV JSON, analyse sémantique, streamlit, python

II. Introduction

Les entreprises et les agences de recrutement traitent quotidiennement de nombreux CV. C'est une tâche chronophage. Un système intelligent est nécessaire pour extraire toutes les informations impotantes des CV non structurés et les transformer tous en un format structuré commun qui peut ensuite être classé pour un poste spécifique. Les informations analysées incluent le nom, l'adresse e-mail, les profils sociaux, les sites Web personnels, les années d'expérience professionnelle, les années d'études, les expériences éducatives, les publications, les certifications, les expériences de bénévolat, les mots-clés et enfin le groupe du CV (ex : informatique, ressources humaines…). Les informations analysées sont ensuite stockées dans une base de données (SQL dans ce cas) pour une utilisation ultérieure. Contrairement à d'autres données non structurées (ex : corps de l'e-mail, contenu de la page Web, etc.), les CV sont un peu structurés. Les informations sont stockées dans des ensembles discrets. Chaque ensemble contient des données sur le contact, l'expérience professionnelle ou les détails de l'éducation de la personne. Malgré cela, les CV sont difficiles à analyser. En effet, ils varient selon les types d'informations, leur ordre, leur style d'écriture, etc. De plus, ils peuvent être rédigés dans différents formats. Certains des plus courants incluent '.txt', '.pdf', '.doc', '.docx', '.odt', '.rtf' etc. Pour analyser les données de différents types de CV de manière efficace et efficiente, le modèle ne doit pas dépendre de l'ordre ou du type de données.

III. Etat de l'art

Sur le marché il existe plusieurs analyseurs de cv utilisant de nombreuses technologies les uns plus efficaces que d'autres.

Pour en citer quelques-uns nous avons Cvparser qui est un analyseur de cv performant a 86%

Nous avons India qui l'un des analyseurs les plus avancé et très performant.

Vu l'engouement autour des algorithmes de traitement de langages naturel plusieurs articles en parlent donc nous nous sommes inspiré pour notre projet.la plupart des articles traitent du sujet en long et en large, nous avons pris le soin de mettre leurs liens dans la partie bibliographie

IV. PREPROCESSING

Le prétraitement des données est la première étape du traitement du langage naturel. Le prétraitement des données est une technique d'exploration de données qui transforme les données brutes en un format compréhensible. Les données du monde réel sont pour la plupart inadéquates, contradictoires et contiennent d'innombrables erreurs. La méthode de prétraitement des données s'est avérée efficace pour résoudre ces problèmes. Le prétraitement des données traite ainsi davantage les données brutes. Les données sont amenées à passer par une série d'étapes au moment du prétraitement

Nettoyage des données : les processus, tels que le remplissage des valeurs manquantes, le lissage des données bruyantes ou la résolution des incohérences, nettoient les données. **Intégration des données** : les données constituées de diverses représentations sont regroupées et les conflits entre les données sont pris en charge.

Transformation des données : les données sont distribuées, assemblées et théorisées.

Réduction des données : l'objectif de cette étape est de présenter un modèle contracté dans un entrepôt de données.

Discrétisation des données : Dans cette étape, le nombre de valeurs d'une caractéristique ininterrompue est réduit par division de la plage d'intervalles de caractéristiques.

V. THE PARSER

Le traitement automatique du Langage Naturel est un des domaines de recherche les plus actifs en science des données actuellement. C'est un domaine à l'intersection du Machine Learning et de la linguistique. Il a pour but d'extraire des informations et une signification d'un contenu textuel.

Le Traitement Automatique du Langage naturel (TAL) ou Natural Language Processing (NLP) en anglais trouve de nombreuses applications dans la vie de tous les jours: traduction de texte

(DeepL par exemple), correcteur orthographique, résumé automatique d'un contenu, synthèse vocale, classification de texte, analyse d'opinion/sentiment, prédiction du prochain mot sur smartphone, extraction des entités nommées depuis un texte.

a. Tokenisation

La tokenisation cherche à transformer un texte en une série de tokens individuels. Dans l'idée, chaque token représente un mot, et identifier des mots semble être une tâche relativement simple. Mais comment gérer en français des exemples tels que: « J'ai froid ». Il faut que le modèle de tokenisation sépare le « J' » comme étant un premier mot.

SpaCy offre une fonctionnalité de tokenisation en utilisant la fonction nlp. Cette fonction est le point d'entrée vers toutes les fonctionnalités de SpaCy. Il sert à représenter le texte sous une forme interprétable par la librairie.

```
Entrée [3]: words = "Fatoumata DIASSANA est une bonne musulmane"
            tokens = [i for i in words.split()]
            print(tokens)

            ['Fatoumata', 'DIASSANA', 'est', 'une', 'bonne', 'musulmane']
```

b. Tokenisation par phrases

On peut également appliquer une tokenisation par phrase afin d'identifier les différentes phrases d'un texte. Cette étape peut à nouveau sembler facile, puisque a priori, il suffit de couper chaque phrase lorsqu'un point est rencontré (ou un point d'exclamation ou d'interrogation).

```
Entrée [6]: def return_token_sent(sentence):
                # Tokeniser la phrase
                doc = nlp(sentence)
                # Retourner le texte de chaque phrase
                return [X.text for X in doc.sents]

            return_token_sent("Ali est riche.Il est avare ")

Out[6]: ['Ali est riche.', 'Il est avare']
```

c. Stemming

Le stemming consiste à réduire un mot dans sa forme « racine ». Le but du stemming est de regrouper de nombreuses variantes d'un mot comme un seul et même mot. Par exemple, une fois que l'on applique un stemming sur « Chiens » ou « Chien », le mot résultant est le même. Cela permet notamment de réduire la taille du vocabulaire dans les approches de type sac de mots ou Tf-IdF.

Un des stemmers les plus connus est le Snowball Stemmer. Ce stemmer est disponible en français.

```
Entrée [7]: test = "Bouygues a eu une coupure de réseau à Marseille"
```

```
Entrée [8]: from nltk.stem.snowball import SnowballStemmer
            stemmer = SnowballStemmer(language='french')

            def return_stem(sentence):
                doc = nlp(sentence)
                return [stemmer.stem(X.text) for X in doc]
```

```
Entrée [9]: return_stem(test)
Out[9]: ['bouygu', 'a', 'eu', 'une', 'coupur', 'de', 'réseau', 'à', 'marseil']
```

d. Reconnaissance d'entités nommées (NER)

En traitement automatique du langage, la reconnaissance d'entités nommées cherche à détecter les entités telles que des personnes, des entreprises ou des lieux dans un texte. Cela s'effectue très facilement avec SpaCy.

```
Entrée [10]: def return_NER(sentence):
                 # Tokeniser la phrase
                 doc = nlp(sentence)
                 # Retourner le texte et le label pour chaque entité
                 return [(X.text, X.label_) for X in doc.ents]

             sentence = "Pierre est née en France"
             # return_NER(sentence)
             from spacy import displacy

             doc = nlp(sentence)
             displacy.render(doc, style="ent", jupyter=True)
```

Pierre PERSON est née en France GPE

e. L'étiquetage morpho-syntaxique

L'étiquetage morpho-syntaxique ou Part-of-Speech (POS) Tagging en anglais essaye d'attribuer une étiquette à chaque mot d'une phrase mentionnant la fonctionnalité grammaticale d'un mot (Nom propre, adjectif, déterminant…).

```
Entrée [20]: def return_POS(sentence):
                 # Tokeniser la phrase
                 doc = nlp(sentence)
                 # Retourner les étiquettes de chaque token
                 return [(X, X.pos_) for X in doc]

             return_POS(sentence)
             # doc = nlp(sentence)
             # displacy.serve(doc, style="dep")

Out[20]: [(Pierre, 'PROPN'),
          (est, 'VERB'),
          (née, 'VERB'),
          (en, 'ADP'),
          (France, 'PROPN')]
```


f. Embedding par mot

Avec SpaCy, on peut facilement récupérer le vecteur correspondant à chaque mot une fois passé dans le modèle pré-entraîné en français.

Cela nous sert à représenter chaque mot comme étant un vecteur de taille 96.

```
Entrée [22]: def return_word_embedding(sentence):
                 # Tokeniser la phrase
                 doc = nlp(sentence)
                 # Retourner le vecteur lié à chaque token
                 return [(X.vector) for X in doc]

             return_word_embedding(test)

Out[22]: [array([ 2.1601634e+00,  4.8973560e-01,  1.1997707e+00, -1.8521857e+00,
                 -2.2568941e+00,  9.0295118e-01,  8.2155323e-01,  4.7497907e+00,
                  2.6569142e+00,  1.7785255e+00,  1.4244055e+00,  9.3683016e-01,
                 -1.9908130e-01, -1.4341477e+00,  2.5288792e+00, -1.2025466e+00,
                  1.0856124e+00,  2.8156676e+00, -4.5342946e-01, -2.7789593e-01,
                  1.3699846e+00,  1.9798943e-01,  2.0850458e+00, -6.2474865e-01,
                  6.0665339e-02,  2.4951179e+00, -7.6523566e-01, -2.2915242e+00,
                 -3.4559162e+00, -2.4522109e+00,  2.7074471e+00, -1.5845439e+00,
                 -3.0513155e+00, -1.4339850e+00,  2.1288407e+00, -4.5562685e-03,
```


g. Similarité entre phrases

Afin de déterminer la similarité entre deux phrases, nous allons opter pour une méthode très simple :

- déterminer l'embedding moyen d'une phrase en moyennant l'embedding de tous les mots de la phrase
- calculer la distance entre deux phrases par simple distance euclidienne

Cela s'effectue très facilement avec SpaCy !

Entrée [21]:
```
import numpy as np
def return_mean_embedding(sentence):
    # Tokeniser la phrase
    doc = nlp(sentence)
    # Retourner la moyenne des vecteurs pour chaque phrase
    return np.mean([(X.vector) for X in doc], axis=0)
test = "Bouygues a eu une coupure de réseau à Marseille"
test_2 = "Le réseau sera bientot rétabli à Marseille"
test_3 = "La panne réseau affecte plusieurs utilisateurs de l'opérateur"
test_4 = "Il fait 18 degrés ici"
np.linalg.norm(return_tensor(test)-return_tensor(test_2))
np.linalg.norm(return_tensor(test)-return_tensor(test_3))
np.linalg.norm(return_tensor(test)-return_tensor(test_4))
```



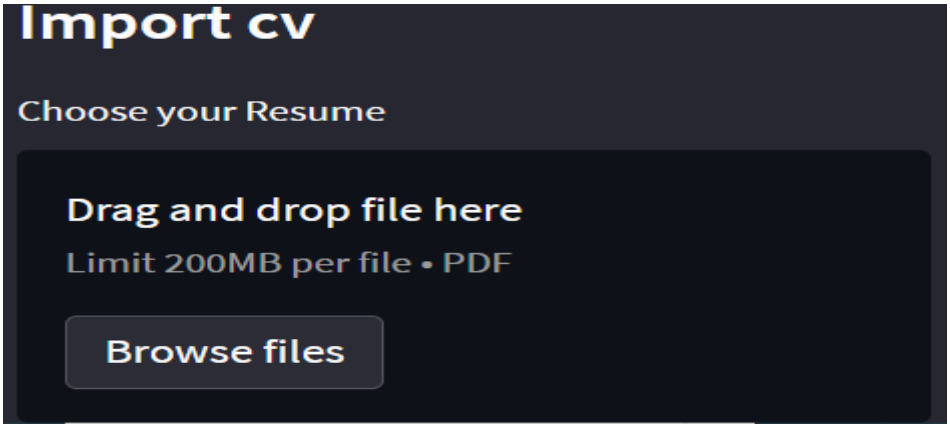
```
16.104986
17.035103
22.039303
```

La phrase 2 est bien identifiée comme la plus proche, puis la phrase 3 et 4. On peut également utiliser cette approche pour classifier si une phrase appartient à une classe ou à une autre.

VI. Présentation de l'application

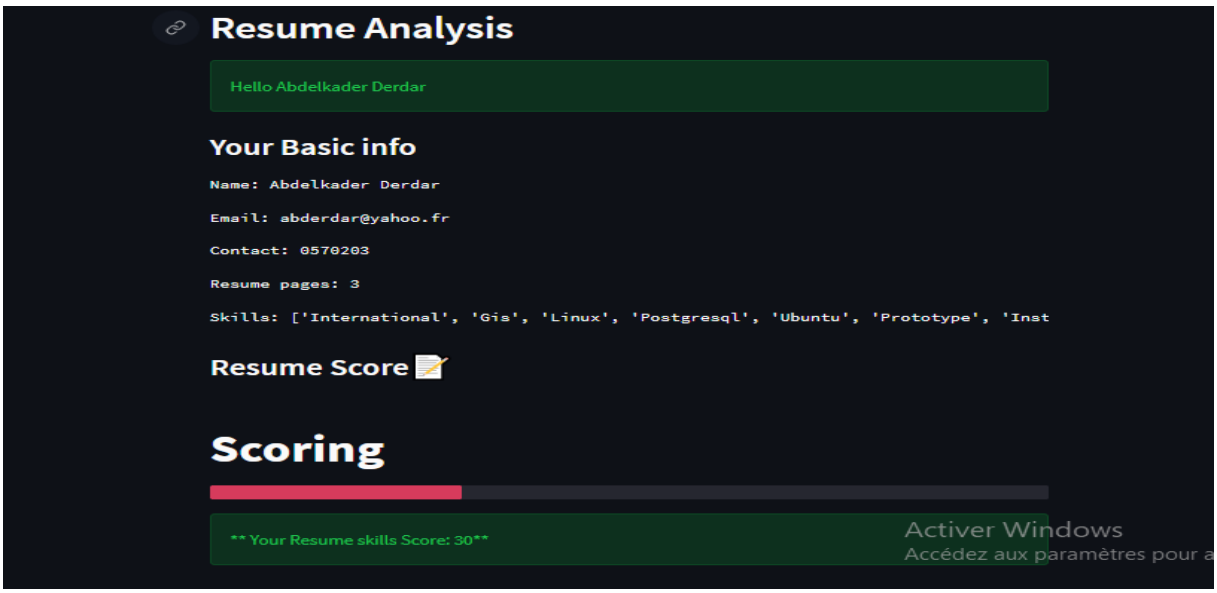
a. Cv uploading

L'utilisateur doit uploader le cv a analyser sous format pdf



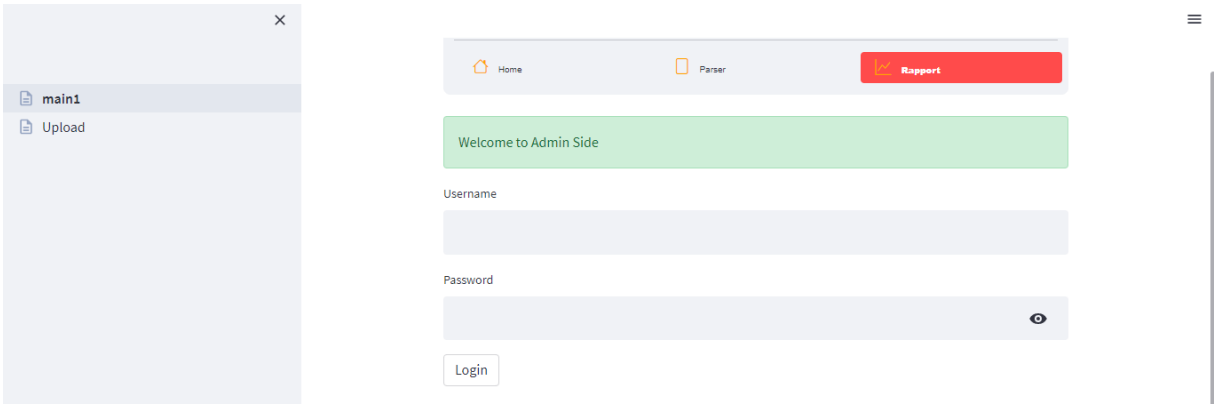

Chose Skills
Php
react
django
Java
Python
Lean
Html
Database
Photoshop


Chose Skills
Php
react
django
Java
Python
Lean
Html
Database
Photoshop
Import cv
Choose your Resume
Drag and drop file here
Limit 200MB per file • PDF
Browse files
383145304-cv.pdf
63.6KB
Home
Parser
Rapport
Prediction
About
1
sur 3
Abdelkader
Derdar
Cité des Annasseurs 4, BT 78, N6,
Kouba Alger
Phone: 057020350
E-mail: abderdar@yahoo.fr
CURRICULUM VITAE
PROFILE
• Ingénieur actif et autodidacte, intéressé par la conception et le développement des Systèmes d'informations et les Bases de Données.
ETUDES ET FORMATIONS
Cycle Superieur
2000-2003 • Université de Amar Tlidji (Laghouat) - niveau 2eme année tronc commun technologie.
2003-2007 • Institut International de Management (INSIM) Hydra, Alger.
• Diplôme Technicien Supérieur en Informatique de Gestion du collège de Sherbrooke.
• Diplôme d'Ingénieur en Informatique de l'école Paris Graduate School of Management (PGSM).
2007-2009 • Etablissement des Techniques Modernes Ibn Rochd.
• Formation Master professionnel en management des Systèmes d'information et des Bases de Données, et Diplôme de concepteur de systèmes d'information de L'institut d'ingénierie informatique de Limoges.
20016 • Formation a l'IFEG: "installateur de système photovolotaic standalone"

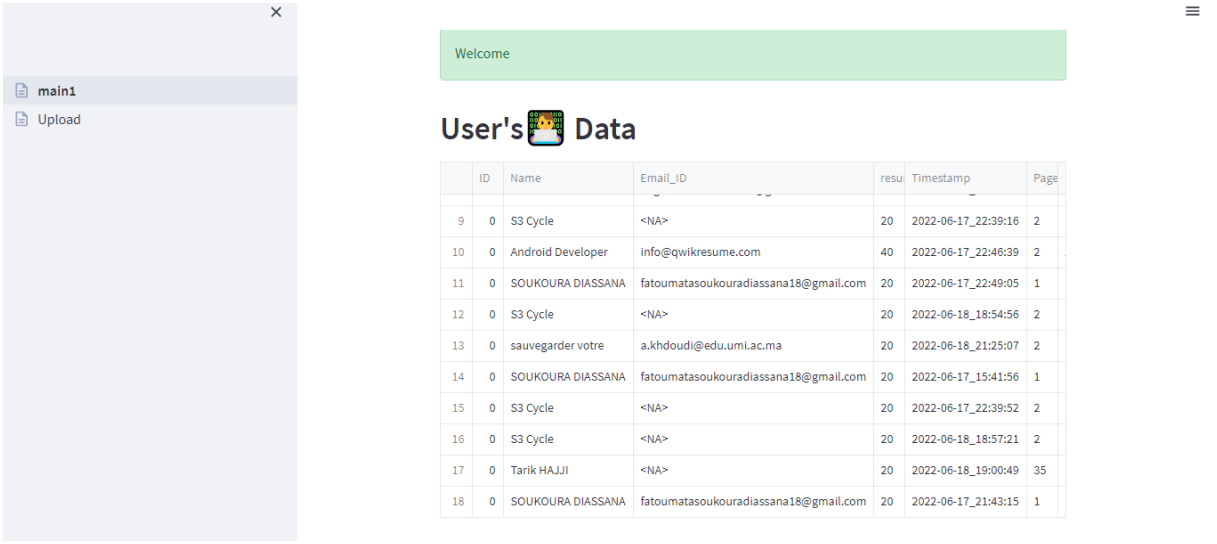

Resume Analysis
Hello Abdelkader Derdar
Your Basic info
Name: Abdelkader Derdar
Email: abderdar@yahoo.fr
Contact: 0570203
Resume pages: 3
Skills: ['International', 'Gis', 'Linux', 'Postgresql', 'Ubuntu', 'Prototype', 'Inst
Resume Score
Scoring
** Your Resume skills Score: 30**

b.Rapport

Identification administrateur

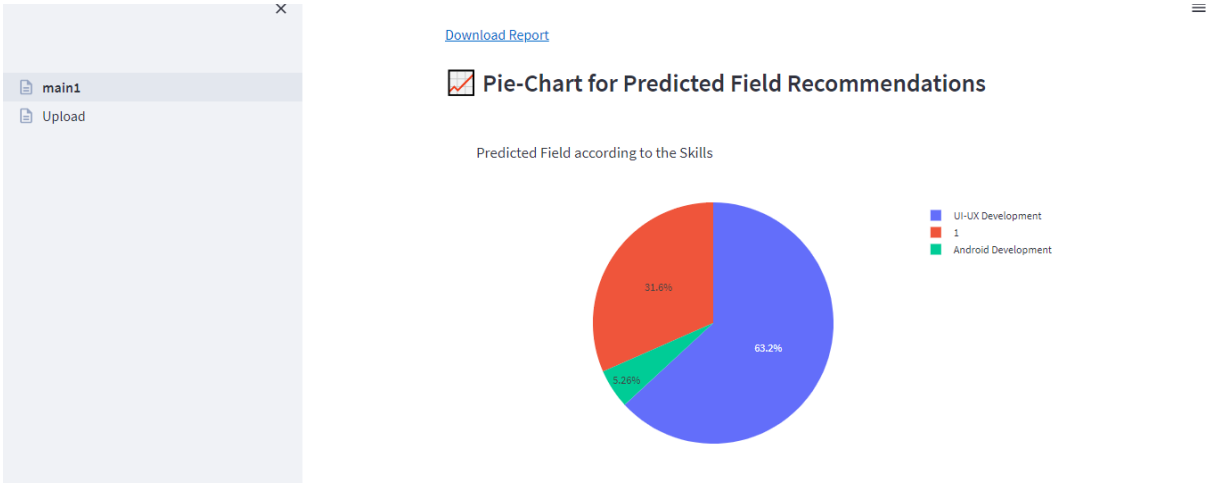


La base de données



	ID	Name	Email_ID	resu	Timestamp	Page
9	0	S3 Cycle	<NA>	20	2022-06-17_22:39:16	2
10	0	Android Developer	info@qwikresume.com	40	2022-06-17_22:46:39	2
11	0	SOUKOURA DIASSANA	fatoumatasoukouradiassana18@gmail.com	20	2022-06-17_22:49:05	1
12	0	S3 Cycle	<NA>	20	2022-06-18_18:54:56	2
13	0	sauvegarder votre	a.khdoudi@edu.umi.ac.ma	20	2022-06-18_21:25:07	2
14	0	SOUKOURA DIASSANA	fatoumatasoukouradiassana18@gmail.com	20	2022-06-17_15:41:56	1
15	0	S3 Cycle	<NA>	20	2022-06-17_22:39:52	2
16	0	S3 Cycle	<NA>	20	2022-06-18_18:57:21	2
17	0	Tarik HAJJI	<NA>	20	2022-06-18_19:00:49	35
18	0	SOUKOURA DIASSANA	fatoumatasoukouradiassana18@gmail.com	20	2022-06-17_21:43:15	1

Repartission des données



Conclusion

Les analyseurs de cv sont une aubaine pour de nombreuses entreprises, ils réduisent le travail chronophage et permettent aux RH de se concentré sur d'autres activités a valeur ajouté. Cependant ils présentent quelques risques.

Références

article

https://blog.apilayer.com/build-your-own-resume-parser-using-python-and-nlp/

pyresparser

https://pypi.org/project/pyresparser/

article

https://intelligence-artificielle.com/selectionner-cv-ia-guide-complet/#:~:text=En%20s'appuyant%20sur%20l,%C3%A0%20trouver%20les%20meilleurs%20candidats.&text=Plusieurs%20entreprises%20ont%20d%C3%A9j%C3%A0%20adopt%C3%A9,acc%C3%A9l%C3%A9rer%20les%20processus%20de%20recrutement.

Article

https://datapeaker.com/big-data/aprenda-analytics-business-analytics-comunidad-de-analytics/

india

https://inda.ai/en/cv-parsing/