DeNp Kongru

Ein NLP-Projekt zur Bestimmung von Kongruenz in deutschen Nominalphrasen in Lernertexten aus dem Lernerkorporus Merlin. Dieses Projekt wurde im Rahmen des computerlinguistischen Kurses `Korpuslinguistische Analysen der Nominalflexion im Deutschen (050041-SoSe23)` an der Ruhr-Universitaet Bochum in Deutschland entwickelt.

Fehler melden · Funktion anfragen

▶ Inhaltsverzeichnis

Ueber das Projekt

Hintergrund

Im Rahmen dieses Projekts soll die Kongruenz in deutschen Nominalphrasen bestimmt werden. Daraus resultierend werden die Ergebnisse aus verscheidenen Lernergruppen miteinander verglichen, um die Unterschiede zwischen den jeweiligen Sprechergruppen z.B. Spanisch, Franzoesisch, Deutsch, etc veranschaulichen zu koennen.

Ergebnisse

Nominale Kongruenz im Deutschen bereitet Lernern der deutschen Sprache etliche Schwierigkeiten aufgrund der Komplexität des deutschen Kasussystems. Aufgrund dessen wurden einfache heuristische Methoden entwickelt, um die Kongruenz innerhalb der Nominalphrasen zu bestimmen. Um diese Problematik besser darstellen zu können, wurden Anglophonen und Frankophonen in Bezug auf ihre Deutschkenntnisse miteinander verglichen. Es wird angenommen, dass Frankophonen weniger Probleme mit Nominalphrasen als Anglophonen hätten, da das Kongruenzprinzip im Französischen stärker vertreten ist als im Englischen. Um dieser Annahme auf den Grund zu gehen, wurden entsprechende Nominalphrasen aus dem Lernerkorpus Merlin extrahiert und analysiert. Die hier erzielten Ergebnisse bezüglich der Eingangshypothese sind nicht eindeutig und somit lässt sich nicht sagen, ob die Hypothese anzunehmen bzw. zu verwerfen ist, wobei die Ergebnisse leicht andeuten, dass beide Sprechergruppen fast gleich stark sind, was Kongruenz im Deutschen betrifft. Man kann jedoch anhand der Ergebnisse sehen, dass um so ein Projekt durchzuführen, könnte es von Vorteil sein, ein Korpus zu analysieren, das für so eine Aufgabestellung angefertigt wurde, denn so könnte man aussagekräftigere Ergebnisse erzielen.

Class	Precision	Recall	F1-Score	Support
0 - EINFACH	0.91	0.50	0.65	20
01 - ART	0.13	0.09	0.11	32
02 - PREP	1.00	0.08	0.15	24
03 - Eigennamen	0.38	0.38	0.38	26
04 - Redewendung bzw. Satz	1.00	0.25	0.40	8
10 - EINFACH (Nicht Kongruenz)	0.33	1.00	0.49	17
11 - ART (Nicht Kongruenz)	0.00	0.00	0.00	9
12 - PREP (Nicht Kongruenz)	0.75	0.35	0.48	17
99 - Unbekannt (Nicht Kongruenz)	0.48	0.77	0.59	48
Accuracy			0.43	201
Macro Avg	0.55	0.38	0.36	201
Weighted Avg	0.53	0.43	0.39	201

Ergebnisse anhand von Sci-Kit learn generiert.

(Zurueck zum Anfang)

Verwendete Ressourcen

Eine Liste der Ressourcen, die bei der Entwicklung des Programms verwendet wurden.

Libraries

Pip

- click==8.1.3
- pandas==1.5.2
- PyYAML==6.0.1
- rich~=12.6.0
- scikit_learn==1.1.3
- textdistance==4.5.0
- tqdm==4.64.1
- typer~=0.4.2

Korpora

- MERLIN Corpus
- Deutsche morphologische Woerterbuecher

Zusaetzliche Tools

Diese Tools wurden benutzt, um die Lernertexte zu taggen und zu parsen. Dieses Verfahren wurde in einem anderen, verwandten Projekt durchgefuehrt.

- Parzu - deutscher Parser
- Conluu - CoNLL-U Parser
    - Wie genau diese Tools verwendet wurden, bitte hier nachlesen

(Zurueck zum Anfang)

Erste Schritte

Korpora entpacken

Das Projekt bezieht sich auf einige Korpora und Datenbanken. Ohne diese kann das Projekt nicht gestartet werden.

Alle Korpora muessen erstmal entpackt werden:

app_resources/data/demorphy

- demorphy_de_kongru.zip

app_resources/data/merlin_corpus

- merlin_corpus.db.zip

Die Verzeichnisse sollte nachher die folgenden Strukturen haben:

Demorphy

- Inhalt
    - demoprhy_dict.pkl
    - demorpy_dict.txt

Merlin

- Inhalt
    - merlin_corpus.db.zip
    - merlin_raw_corpus.zip
        - Das sind die Datein, womit die SQL-DB erzeugt wurde. Diese muss man nur entpacken
        - wenn man eine neue SQL-DB erstellen moechte.
    - CHANGELOG.md
    - README.md
    - merlin_corpus.db

Andere Korpora und Dateien sind zwar enthalten, aber diese muessen nicht zwangslaeufig entpackt werden.

(Zurueck zum Anfang)

Hauptverzeichnis festlegen

Als naechstes muss das Hauptverzeichnes des Projekts festgelegt werden. Dies tut man in der `main_config.yaml` Datei. z.B.

- `/Users/christopherchandler/de_np_kongru`

Wenn das Hauptverzeichnis nicht richtig gesetzt wurde, kann das Programm nicht wie erwartet gestartet werden!

Voraussetzungen

Das Programm wurde mit Python 3.10 konzepiert und entwicklet. Es besteht die Moeglichkeit eine altere Python-Version zu benutzen, aber dann kann es sein, dass das Programm nicht stabil ist.

(Zurueck zum Anfang)

Installation

Um die notwendige libraries installieren zu koennen, das folgendene Kommando in der Konsole ausgeben: `pip install -r requirements.txt`

(Zurueck zum Anfang)

Anwendung

API

Wenn man die Module einfach so importieren will, kann man das auch ueber den ganz normalen Weg machen.

```
from kongru.api_nlp.congruential_analysis.app_congruential_analysis import
(
 nominal_phrase_agreement_analysis )
```

(Zurueck zum Anfang)

CLI

Um DeNP Kongru als CLI starten zu koennen, den `python main.py` im Hauptverzeichnis ausfuehren. Wenn alles korrekt eingerichtet wurde, soll Folgendes in der Konsole erscheinen:

```
Usage: main.py [OPTIONS] COMMAND [ARGS]...

  Die Hauptapp von DeNpKongru

Commands:
  verzeichnis_leeren  Ein ausgewaehltes Verzeichnis leeren
  kongruenz_leeren    verzeichnis_leeren
```

```
  kongruenz           Die Np-Eintraege auswerten
  datenbank           Die Datenbankcorpora verwalten und durchsuchen
  statistik           Eine einfache Analyse ueber DeNpKongru ausfuehren
```

Bei jedem Befehl kann man einen Hinweis ausgeben lassen, wie die Befehle funktionieren und welche Argumente erforderlich sind, indem man `--help` am Ende eingibt.

Z.B.: `python main.py datenbank text_lesen --help`

```
Usage: main.py datenbank text_lesen [OPTIONS]

  einen bestimmten Text in der Datenbank lesen

Options:
  --text_id, --id TEXT  Die Text-Id des gewuenschten Textes angeben
[default:
                        1031_0003130]
  --help                Show this message and exit.
```

Um genauer zu wissen, wie diese Schnittstelle funktioniert oder Fehlermeldung besser verstehen zu koennen, bitte die Dokumentation von Typer durchlesen.

(Zurueck zum Anfang)

Ordnerstruktur

Die Dateien, die DeNpKongru braucht, um eine Analyse durchzufuehren.

Die Order sind zwar leer, aber werden befuellt, je nachdem welche Kommandos man ausfuehrt. Man kann auch die Dateien manuell in dem entsprechenden Ordner ablegen.

- user
  - Die dateien, die vom Benutzer abgelegt, generiert oder benutzt wird.
    - incoming
      - ast
        - Die Conll-Datei als Python Listen
      - conll
        - Die Conll-Dateien
      - full_json
        - Die Merlin-Texte als Json-Dateien
      - pylist
        - Das Gleiche wie die AST-Dateien, aber die interne Struktur ist eine andere.
      - raw
        - Die einfachen Merlin-Texte
    - kongru_evaluation
      - Die Dateien, um das Programm auszuwerten, werden hier gespeichert
        - gold_files
          - alle korrigierter NP-Dateien muessen in diesem Verzeichnis liegen

        - raw_files
            - die entsprechenden unkorrigierten Dateien muessen auch in diesem Verzeichnis liegen. Sie werden miteinander verglichen, um *precision*, *recall* und *f1-score* zu generieren.
    - outgoing
      Hier werden die Ergebnisse gespeichert.
        - batch_results
            - wenn man multi_np_analysis.py ausfuehrt, werden die Ergebnisse hier als .csv-Datei gespeichert.
        - extracted_nominal_phrases
            - Die unverarbeiteten Nominalphrasen, die aus den AST oder Pylist-dateien extrahiert wurden, werden hier gespeichert.
            - nominal_phrase_analysis_csv_results
            - Die Auswertung der Nominalphrasen werden hier gespeichert.
        - nominal_phrase_analysis_json_results
            - Die Nominalphrasen und deren Ergebnisse werden in der entsprechenden JSON-Datei gespeichert.
        - text_ids
            - Dieser Ordner enthaelt die Dateien `test_ids.txt` und `training_ids.txt`. Die sind wichtig fuer multi_np_analysis.py . Alle Ids, sofern sie in der SQL-DB existieren, werden eingelesen und ausgewertet.

(Zurueck zum Anfang)

Analyse durchfuehren

Die Analyse kann man entweder ueber die API oder die CLI durchfuehren. An run_main_analysis.sh kann man sich orientieren, wenn man neue Skripte anlegen moechte.

Wenn die Pfade korrekt eingerichtet wurden, sollte dieses Skript problemlos funktioniern. Man muss also nur das Skript starten oder es durch die CLI aufrufen.

In dieser Skript-datei wird `main.py kongruenz multi` bzw. `multi_nominal_phrase_agreement_analysis` ausgefueht. Hier werden meherere Text-Id auf meherere Dateien analysiert (Batch Analysis).

Es bleibt einem ueberlassen, wie und wo man dieses Skript ausfuehrt.

(Zurueck zum Anfang)

Nominalphrasen - Dateien

Das Programm erwartet die folgende Struktur:

CSV - Eingangsdatei

Es muss eine .CSV Datei mit der folgenden Struktur eingegeben werden:

```
NP_ID, Nominalphrase,Morphologische Information, Satz
1_1,Maria Schmidt,Maria N Masc|_|Sg,Schmidt N Masc|_|Sg,Maria Schmidt
```

Alle Ergebnisse Dateien haben die folgene Struktur:

CSV - Ergebnisdatei

Die Ergebnisse werden auch in einer separaten CSV-Datei gespeichert, damit man die Ergebnisse auf den ersten Blick verstehen kann.

```
CSV
NP_ID,Ergebniscode, Nominalphrase, Morphologische Information, Satz
1_2,3,einem Haus,"einem,ART,Indef|Neut|Dat|Sg","Haus,N,_|Neut|Dat|Sg","
einem Haus suchen ."
```

JSON - Ergebnisdatei

Die Ergebnisse werden auch in einer separaten Json-Datei gespeichert, damit sie in einem anderen Programm weiter verarbeitet werden koennen.

Die .json hat dann die folgende Struktur.

```
 {
        "file_ID": "1023_0101841",
        "sent_NP_ID": "1_1",
        "sentence": "Maria Schmidt Addresse Computer Spezialist
Odenwaldstra\u00dfe 5.",
        "np_congruency_info": {
            "congruency_code": "1",
            "nominal_phrase": "Maria Schmidt"
        },
        "metadata": {
            "corpus": "MERLIN_DE",
            "author": {
                "author_ID": "1023_0101841",
                "L1": "English",
                "age": "32",
                "gender": "female"
            },
            "CEFR": {
                "overall_fairRating": "B1+",
                "test_level": "B2"
            },
            "task": {
                "formality": "formal",
                "text_type": "letter",
                "topic": "apply for internship in sales department"
```

```
            }
        },
```

(Zurueck zum Anfang)

Ergebniscodes

Es werden hier verschiedene Kongruenzcodes aufgelistet, um festzustellen, um welche Art von Kongruenz es sich handelt.

Arten	Beispiel
EINFACH	Stadt
ART	Das Leben
PREP	Mit Kindern
Eigennamen	Katharina, Maria Meier
Redewendungen	Liebe Julia, Mit freundlichen Grüßen

Beispiele

- `EINFACH` - Nomen, die alleine bzw. ohne Artikel vorkommen.
- `ART` - Nomen, die mit Artikeln oder Adjektiven vorkommen
    - Wenn es nicht kongruiert, liegt es vermutlich daran, dass es Rechtschreibfehler vorliegen
- `EIGENNAMEN` - Sie sind immer richtig. Das sind z.B. Namen oder Städte.
- `Redewendung` - Sofern sie keine Rechtschreibfehler aufweisen, kann man davon ausgehen, dass sie immer richtig sind. Das sind wie z.B.
    - Mit freundlichen Grüßen
    - Sehr geehrte Damen und Herren
    - Liebe(r) Frau/Herr Schmidt

Wenn eine Np in einer Kategorie vorkommt, kann man erkennen, ob es kongruiert oder nicht und warum.

z.B.

```
10 – Es ist eine Einfache Nominalphrase,
aber es liegen Rechtschreibfehler vor
```

(Zurueck zum Anfang)

Kongruenz

Gruende, weswegen die Nominalphrase kongruiert

Code	Bedeutung

Code	Bedeutung
0	EINFACH
1	ART
2	PREP
3	Eigennamen
4	Redewendung bzw. gaengiger Satz

(Zurueck zum Anfang)

Nicht Kongruenz

Gruende, weswegen die Nominalphrase nicht kongruiert.

Code	Bedeutung
10	EINFACH
11	ART
12	PREP
99	Unbekannt

- 99
  - Aus unbekannten Gruenden konnte die Kongruenz nicht bestimmt werden. Es ist vermutlich irgendwo ein Fehler aufgetreten. Es kann auch sein, dass die Kongruenz einfach nicht ermittelt werden konnte.

(Zurueck zum Anfang)

Roadmap

Siehe die offenen Probleme fuer eine vollstaendige Liste der vorgeschlagenen Funktionen (und bekannten Probleme).

(Zurueck zum Anfang)

Beitragen

Beitraege sind es, die die Open-Source-Community zu einem erstaunlichen Ort zum Lernen, Inspirieren und Erschaffen machen. Jegliche Beitraege, die Sie leisten, werden **sehr geschaetzt**.

Wenn Sie eine Idee haben, die dieses Projekt verbessern wuerde, bitte machen Sie einen Fork des Repositories und erstellen Sie einen Pull Request. Sie koennen auch einfach ein Problem mit dem Tag "Verbesserung" eroeffnen. Vergessen Sie nicht, dem Projekt einen Stern zu geben! Vielen Dank nochmals!

1. Forken Sie das Projekt.
2. Erstellen Sie Ihren Feature-Branch (`git checkout -b feature/ErstaunlicheFunktion`).
3. Machen Sie Ihre aenderungen (`git commit -m 'Fuege einige erstaunliche Funktionen hinzu'`).
4. Pushen Sie den Branch (`git push origin feature/ErstaunlicheFunktion`).
5. Eroeffnen Sie einen Pull Request.

(Zurueck zum Anfang)

Lizenz

Vertrieben unter der MIT-Lizenz. Siehe `LIZENZ` fuer weitere Informationen.

(Zurueck zum Anfang)

Kontakt

Christopher Chandler - christopher.chandler at outlook.de

- Project Link: De_NP_Kongru

(Zurueck zum Anfang)

Danksagungen

- Imge Yuezuencueoglu
- Georg Stin
- Ikram Abdalla

(Zurueck zum Anfang)