Sitzungsberichte der 12. Konferenz über Sprachressourcen und -evaluierung (LREC 2020), S. 3782-3789
Marseille, 11.-16. Mai 2020

OpusTools und Parallel-Corpus-Diagnostik

Mikko Aulamo, Umut Sulubacak, Sami Virpioja, Jorg Tiedemann
Department of Digital HumanitiesUniversity
of Helsinki, Helsinki/Finnland{
mikko.aulamo, umut.sulubacak, sami.virpioja, jorg.tiedemann}@helsinki.fi

Zusammenfassung

Dieses Papier stellt OpusTools vor, ein Paket zum Herunterladen und Verarbeiten von Parallelkorpus, das in der OPUS corpus-Sammlung enthalten ist.Das Paket implementiert Werkzeuge für den Zugriff auf komprimierte Daten in ihrem archivierten Release-Format und ermöglicht es, einfach zwischen gängigen Formaten zu konvertieren.OpusTools umfasst auch Werkzeuge zur Sprachidentifizierung und Datenfilterung sowie Werkzeuge zum Importieren von Daten aus verschiedenen Quellen in das OPUS-Format.Wir zeigen den Einsatz dieser Werkzeuge in paralleler Korpus-Erstellung und Datendiagnose.Letzteres ist besonders nützlich für die Ermittlung potenzieller Probleme und Fehler im umfangreichen Datensatz.Mit diesen Tools können wir nun die Gültigkeit von Datensätzen überwachen und die Gesamtqualität und Konsistenz der Datenerhebung verbessern.

Schlagwörter: Korpus (Erstellung, Annotation usw.);Maschinelle Übersetzung;Werkzeuge, Systeme, Anwendungen

1. Einleitung

Opus (Tiedemann, 2012) ist die größte Sammlung offen verfügbarer Parallelkorpus.Die Sammlung ist im Laufe der Jahre stetig gewachsen und wird in der Arbeit an maschineller Übersetzung und linguistischer Forschung weit verbreitet.Derzeit enthält es 57 freigegebene Korpus für über 700 Sprachen und Sprachvarianten, die mehr als 70.000 Bitexte im Sinne von ausgerichteten Sprachpaaren über alle Korpusin der Sammlung erzeugen.Die Größe und Popularität von OPUS macht es notwendig, eine effiziente Infrastruktur zu bauen, die es den verschiedenen Benutzern ermöglicht, die Daten zu erhalten und zuzugreifen, und dieses Papier stellt zwei Pakete vor, die Werkzeuge für diesen Zweck zur Verfügung stellen.Das Ziel dieser Pakete ist es, das Herunterladen, Konvertieren und Verarbeiten der in OPUS enthaltenen Daten von der Kommandozeile oder von Anwendungen mit Hilfe der Bibliothek, die diese Tools implementiert, zu erleichtern.Die beiden Pakete beziehen sich-auf eine Python-Bibliothek mit Kommandozeilen-Tools und einem ergänzenden Perl—Modul, sowohl als Open Source als auch mit permissiven Lizenzen.

In den folgenden Abschnitten stellen wir die Tools und deren grundlegende Verwendung vor und diskutieren auch, wie wir diese Tools zur Erstellung neuer Datensätze und zur systematischen Diagnostik der gesamten Datenbank eingesetzt haben.Mit der Verfügbarkeit der OpusTools ist es nun möglich, die umfangreichen Datensätze sorgfältig zu überprüfen, um die Gültigkeit der Kodierung zu überprüfen, defekte Links und Strukturen zu finden und andere Probleme mit den Daten zu identifizieren.

2. Merkmale von OPUS

Opus enthält parallele Korpus aus einer Vielzahl von Quellen.Jede von ihnen kommt mit ihren eigenen Besonderheitenund die Eigenschaften können erheblich variieren, je nach den ursprünglichen Daten und deren Verteilung.Die Philosophie von OPUS besteht darin, Markup und Annotation so weit wie möglich zu halten, aber das wesentliche Datenformat zu vereinheitlichen, um den Zugriff auf parallele Daten so transparent wie möglich zu machen.Das bedeutet, dass Corpus-Daten in eigenständiges (schemafreies) XML umgewandelt werden, das Original-Markup behält, aber konsequent wesentliches Markup hinzufügt, das für die Ausrichtung und weitere sprachlicheVerarbeitung notwendig ist.Align wird als Standoff-Annotation im XCES Align-Format (für Satzausrichtung) und „Moses-Format“ (für Wortausrichtung) gespeichert.Mit diesem Prinzip können Daten von der Ausrichtungsnotation getrennt gehalten

werden, die eine effiziente Implementierung und Speicherung massiv paralleler Daten ermöglicht und gegebenenfalls auch alternative Ausrichtungen ermöglicht.Abbildung 1 zeigt ein Beispiel für Standoff-Annotation, die in OPUS zur Festlegung von Verbindungen zwischen Sätzen verwendet wird.Jede Satzausrichtungsdateikann eine beliebige Anzahl von LinkGrp— Elementen enthalten, um Dokumente aus einer Datensammlung auszurichten.Dokumente werden mit einem Pfad relativ zur XML-Root des OPUS-Unterkorpus spezifiziert und Linkelemente liefern die Satzausrichtung durch Sätze von Satz-IDs, die durch Semikolon getrennt sind.Das Erstellen einer alternativen Ausrichtung erfolgt einfach durch das Erstellen einer neuenSatzausrichtungsdatei und es müssen keine weiteren Änderungen mit den ursprünglichen Korpusdaten vorgenommen werden.Beachten Sie, dass die Satzausrichtung zweisprachig ist, wie im Beispiel gezeigt.Die Standoff-Annotationermöglicht es jedoch, massiv parallele Datensätze über alle Sprachpaare hinweg auszurichten, ohne eine der verknüpften Datendateien zu duplizieren.Darüber hinaus kann es alternative Korpusdateien mit unterschiedlichen Annotationsebenen geben, ohne dass diese alternativen Dateien neu ausgerichtet werden müssen.Abbildung 2 zeigt Beispiele für solche kommentierten Dateien, die alle auf die gleiche Weise mit der in externen Dateien gespeicherten Standoff-Satzausrichtung ausgerichtet sind.Weitere Details zu den Datenstrukturen in OPUS finden Sie im OPUS Wiki.[1]

Ein weiterer Grundsatz in OPUS besteht darin, die Daten in anderen gängigen Formaten bereitzustellen, um sie für eine Vielzahl von Anwendungen leicht zugänglich zu machen.Diese Datenformate werden jedoch nur aus der zugrunde liegenden XML-basierten Kodierung generiert, die als Hauptkopie jedes Korpus dient.Benutzer von OPUS-Daten sind sich dieser Grundsätze in der Regel nicht bewusst und laden das Datenformat herunter, das am meisten ihren Bedürfnissen entspricht.

Die Idee von OpusTools besteht nun darin, den Zugriff auf Stammdaten in XML und auf die anderen generierten Formate zu vereinheitlichen, indemwichtige Bibliotheken und Kommandozeilenwerkzeuge zum Abrufen und Konvertieren von Korpusdaten zur Verfügung gestellt werden.Sie bieten auch bequeme Werkzeuge für die grundlegende Filterung und den zufälligen Zugriff in archivierten Daten in ihrer komprimierten Form, die für die Verteilung der Daten verwendet wird.Letztere ist besonders wichtig, da die Größe einiger Kor—

[1] Http:// opus.nlpl.eu/trac/wiki/ Datenformate

```
<?xml version=„1.0“ codierung=„utf-8“?>
<!DOCTYPE cesAlign PUBLIC
                    „-//CES//DTD XML cesAlign//EN“ „“> <cesAlign
version=„1.0“>
<linkGrp targType=„s“
          fromDoc=„en/0/1089124/4995691.xml.gz“
          zuDoc=„fr/0/1089124/4588599.xml.gz“>
<link id=„SL0“ xtargets=„1;1“ Überlappung=„0.331“/>
<link id=„SL1“ xtargets=„2 3;2“ Überlappung=„0.560“/> <link id=„SL2“
xtargets=„4;“/>
<link id=„SL3“ xtargets=„5 6;3“ Überlappung=„0.854“/> <link id=„SL4“
xtargets=„7 8 9;4“ Überlappung=„0.699“/> <link id=„SL5“ xtargets=„10
11;5“ Überlappung=„0.776“/>
```

Abbildung 1:Ein Beispiel für Standoff-Satzausrichtung im XCES Align-Format.Das linkGrp —Element gibt die ausgerichteten Dokumentpaare an und die Verknüpfungen zwischen einzelnenSätzen werden in den Linkelementen angegeben .Die optionalen Überlappungsattribute in diesem Beispiel beziehen sich auf Zeitüberlappungsverhältnisse, die als Funktion in der Untertitelausrichtung verwendet werden.

Pora ist so umfangreich, dass es verlangt, dass gemeinsame Dateisysteme die Daten in roher, unkomprimierterForm verarbeiten.So enthält der neueste OpenSubtitles corpus etwa 3,7 Millionen Einzeldokumente in 67 Sprachen mit Ausrichtung in über 3.600 Bitexten.Eine der neuesten Ergänzungen, JW300 deckt 380 Sprachen in über 46.000 Bitexten ab.Insgesamt gibt es über 9,2 Millionen-Einzeldokumente nur in den letzten Veröffentlichungen aller Korpora und diese Zahl wird durch die verschiedenen Vorverarbeitungstypen, die zur Verfügung gestellt werden, Rohtext und tokenisierte Korpora, die teilweise mit zusätzlichen sprachlichen Informationen kommentiert werden, verdoppelt.Darüber hinaus werden Bitexte im nativen XML-Format veröffentlicht(siehe Abbildung 2), Klartextformat und Translation MemoryExchange (TMX).Die Releases belegen derzeit insgesamt 5.9 TB Platz im komprimierten Format.
Die obigen Zahlen verdeutlichen die Notwendigkeit geeigneter Infrastrukturenund effizienter Instrumente zur Verwaltung der verschiedenen Datensätze.Dies ist die Motivation zur Umsetzung der im Folgenden beschriebenen frei verfügbaren OPUS-Tools.Sie erstellen eine komfortable Bibliothekund Werkzeugkasten zum Herunterladen, Extrahieren und Konvertierenvon Daten aus der OPUS-Sammlung.Darüber hinaus helfen sie bei der systematischen Diagnostik der Sammlung, um Fehler und Probleme in den Datensätzen zu identifizieren.Nachfolgend werden wir zunächst die beiden Pakete und deren Funktionalität vorstellen.Anschließend informieren wir über ihre Verwendung bei der Erstellung neuer Datensätze und schließlich berichten wir über die Anwendung von OPUS-Tools für diagnostische Studien und Gesundheitskontrollen.

3. Das OpusTools-Paket

Das OpusTools-Paket ist ein Toolkit zum Herunterladen und Verwalten paralleler Corpora-Daten von OPUS.Das Paket-besteht aus einer Python-Bibliothek und zugehörigen Kommandozeilenskripten.Zusätzlich gibt es ein Perl-Paket für die Erstellung neuer Datensätze und den Zugriff auf parallele Daten.

3.1. Befehlszeilen-Werkzeuge

Das OpusTools-Paket enthält fünf Python 3 basierte Kommandozeilenskripte:Opus_read, opus_express, opus_cat, opus_get und opus_langid.2 Die[Skripte] erlauben das

[2]https: //github.com/Helsinki-NLP/OpusTools

OPUS-Daten, die Ausgabe der Daten in bestimmten Formaten, dasExtrahieren von Trainings-, Entwicklungs- und Testsets aus den Daten und mehr.Abbildung 3 zeigt einen Überblick über die Skripte.

opus_read ist ein Skript zum Herunterladen von Parallelkorpus und zum Konvertieren in gewünschte Formate.Opus corpora enthält XCES-Format-Ausrichtungsdateien, die auf zwei XML-Satzdateien in verschiedenen Sprachen verweisen.Das XCES-Ausrichtungsformat verknüpft die Sätze in Quelldateien mit den Sätzen in Zieldateien mittels Satz-IDs. Die Satzdateien in OPUS-Corpora werden in ZIP-Archive komprimiert und opus_read macht es bequem, die Daten direkt aus den komprimierten Dateien zu lesen. opus_read parsiert eine gegebene Ausrichtungsdatei und erzeugt eine Ausgabe in einem von vier Formaten:normale, moses, TMX- oder XCES-Links. opus_readversucht zunächst die OPUS-Dateien aus lokalen Verzeichnissen zu lesen.Wenn die benötigten Dateien nicht gefunden werden, bietet das Tool eine Möglichkeit, sie herunterzuladen.Die Satzdateien können im rohen, tokenisierten oder geparsierten Format heruntergeladen werden.

opus_read enthält grundlegende Filter zum Entfernen unerwünschter Satzpaare vor dem Erstellen der Ausgabedatei.Nichtausrichtungen, bei denen die Quelle oder das Zielsegment leer ist, können ausgelassen werden.Alternativ kann eine bestimmte Anzahl von Quell- und Zielsegmenten angegeben werden, z. B. ist es möglich, nur Einzel-zu-Eins-Ausrichtung in die Ausgabe aufzunehmen.Einige Corpora enthalten eine Attribut-Score für jedes Satzpaar.Zum Beispiel haben Satzpaare in den offenen Untertiteln corpus überlappende Partituren, die angeben, in welchem Maße sich die Zeitstempel der beiden Segmente überlappen. opus_readkann Satzpaare herausfiltern, die keine vorgegebene Attribut Partiturschwelle überschreiten.Darüber hinaus können Segmentpaareauf Basis von Sprachkennwertwerten entfernt werden.Mit opus_langid—Skript können Sprachbeschriftungen und Selbstvertrauens-Scores zu Satz—XML-Dateien hinzugefügt werden.

opus_express ist ein Skript,das auf opus_read aufgebaut ist undbereite Trainings-, Entwicklungs- und Testsets für ein Sprachpaar aus einem oder mehreren OPUS-Corpora extrahieren kann.Das Verfahren fülltzunächst die vorgegebene Satzquote für den Testsatz aus, setzt dann mit dem gleichen für den Entwicklungssatz fort und wirft den Rest in den Trainingssatz.Das Skript kann wahlweisedie Daten vor dem Aufspalten vorshufflen, oder umgekehrt, markieren und bewahren Dokumentgrenzen über die Aufspaltungen für Dokument-Level-Modelle. opus_express enthält aucheine Option zur Verwendung von Attribut-Scores wie Überlappungswerte, wie sie von opus_read inseinem Qualitäts-Awareness-Toggle extrahiert werden, derhöhere Vertrauens-Satzpaare vorstellt, die eine konfigurierbare Schwelle übersteigen, die in die in die Test- und Entwicklungssätze sortiertwerden soll.

opus_cat wird zum Lesen von einsprachigen Korpus aus OPUS oder einzelnen Dateien innerhalb dieser Korpus verwendet.Die Dateien können im XML-Format ausgedruckt oder in Klartext umgewandelt werden. opus_cat ist nützlich, um die Domäne oder die Qualität eines einzelnen Korpus manuell zu überprüfen, da es in der Lage ist, Dateien direkt aus dem ZIP-Archiv in OPUS corpora zu lesen.

opus_get ist ein Skript zum Herunterladen paralleler Korpusdateien von OPUS.Vor dem Herunterladen kann Korpora nach Name, Quellsprache und Zielsprache gesucht und aufgelistet werden.Zum Beispiel kann man Dateien für ein bestimmtes Sprachpaar in einem einzigen Korpus herunterladen, alle Sprachpaar-Dateien in

Rohes XML-Format:

```
<?xml version=„1.0“ codierung=„utf-8“?>
<document>
  <CHAPTER ID=„1“>
    <P id=„1“>
      <s id=„1“>Wiederaufnahme der Sitzung</s>
    </P>
    <SPEAKER ID=„1“NAME=„Präsident“>
      <P id=„2“>
        <s id=„2“>Ich erkläre die am Donnerstag, den 14. Juni 2001 unterbrochene Sitzungsperiode des Europäischen Parlaments für wiederaufgenommen.</s> </P>
```

Tokenisiertes (angemerktes) XML-Format:

```
<?xml version=„1.0“ codierung=„utf-8“?>
<Dokument><CHAPTER ID=„1“><P id=„1“>
<s id=„1“>
<chunk type=„NP“ id=„c-1“>
  <w hun=„NN“ tree=„NN“ lem=„resumption“ pos=„NN“ id=„w1.1“>Resumption</w>
</chunk>
<chunk type=„PP“ id=„c-2“>
  <w hun=„IN“ Baum=„IN“ lem=„von“ pos=„IN“ id=„w1.2“>von</w>
</chunk>
<chunk type=„NP“ id=„c-3“>
  <w hun=„DT“ tree=„DT“ lem=„the“ pos=„DT“ id=„w1.3“>the</w>
  <w hun=„NN“ tree=„NN“ lem=„session“ pos=„NN“ id=„w1.4“>session</w>
</chunk>
</s>
```

UD Geparstes XML-Format:

```
<?xml version=„1.0“ codierung=„utf-8“?>
<document>
<CHAPTER ID=„1“>
  <P id=„1“>
    <s id=„1“>
      <w xpos=„NOUN“ head=„0“ feats=„Number=Sing“ UPOS=„NOUN“ lemma=„Resumption“ id=„1.1“ deprel=„root“>Resumption</w> <w xpos="ADP" head="1.4"
      UPOS="ADP" lemma="of" id="1.2" deprel="case
      <w xpos=„DET“ head=„1.4“ feats=„Definite=Def|PronType=Art“ UPOS=„DET“ lemma=„the“ id=„1.3“ deprel=„det“>the</w>w xpos=„NOUN“ head=„1.1“
      feats=„Mod=Sing“ UPOS=„NOUN“
    </s>
```

Abbildung 2:Beispiele für XML-kodierte Daten in OPUS.Verschiedene Arten von Anmerkungen können hinzugefügt werden, ohne die Satzausrichtung zu zerstören, die als Standoff-Annotation von Verknüpfungen zwischen Satz-IDs gespeichert wird.

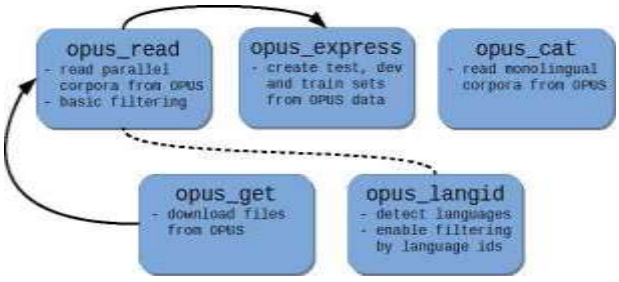


Abbildung 3:Die fünf Python basierten OpusTools Skripte.Jedes Skript kann einzeln verwendet werden. opus_express wirdauf opus_read aufgebaut und opus_readverwendet opus_getzum Download von OPUS-Dateien. opus_langid muss auf Satzdateien angewendetwerden, um dieSprach-ID-Filterung für opus_read zu ermöglichen.

ein einzelner Korpus oder alle Dateien für eine bestimmte Sprache im gesamten OPUS. opus_read verwendet opus_get zum automatischen Herunterladender angeforderten Korpus-Dateien.

opusJangid wird zum Hinzufügen von Sprachkennzeichnungs-Etikettenund Vertrauens-Scores für jeden Satz in einer gegebenen XML-Satzdatei verwendet.Die Sprachidentifizierung erfolgt mit zwei Werkzeugen außerhalb des Regals:Pycld2,[3]die Python-Bindungen für Compact Language Detector 2 und[4] langid.py (Lui und Baldwin, 2012). opus_langid muss auf Satz-XML-Dateien angewendet werden, bevoropus_read Satzpaarenach ihren Sprachbeschriftungen filtern kann .Abbildung 4 zeigt ein Beispiel einer Satzdatei, die mit opus_langid verarbeitet wurde.

3.2. Die OpusTools Python Bibliothek

Neben Kommandozeilen-Skripten werden opus_read, opus_cat , opus_getund opus_langid mit Python-Modulenassoziiert, die in eigenen Skripten importiert und verwendet werden können.Die Module bieten die gleiche Funktionalitätwie die Kommandozeilen-Tools und auch detailliertere Datenverwaltungssteuerung durch den Einsatz von Submodulen und Funktionen.Der gesamte Python-Code ist in Python 3 geschrieben.

OpusRead Modul kann mit Parametern initialisiert werden, die mit den Flags von opus_readübereinstimmen und zum Herunterladen und Konvertieren von Korpusdateien von OPUS verwendet werden.Intern verwendet OpusRead XML —Parsing-Moduleaus der Parse-Unterbibliothek, die im Opus-Tools Python-Paketenthalten ist.Die Unterbibliothek enthält Module zum Parsen von XCES-Alignment- und Satzdateien.Die

[3]https://github.com/aboSamoor/pycld2
[4]https://github.com/CLD2Owners/cld2

```
<?xml version=„1.0“ codierung=„utf-8“?>
<text>
<p id=„1“>
    <s cld2=„en“ cld2conf=„0.99“ id=„s1.1“ LangID=„en“ langidconf=„1.0“>
     Erklärung der Regierungspolitik des Ministerpräsidenten Ingvar Carlsson bei der Eröffnung des schwedischen Parlaments am Dienstag, den 4. Oktober 1988.
    </s>
</p>

<p id=„2“>
    <s cld2=„en“ cld2conf=„0.98“ id=„s2.1“ LangID=„en“ langidconf=„1.0“>
     Eure Majestäten, Eure königlichen Hoheiten, Herr Sprecher, Mitglieder des schwedischen Parlaments.
    </s>
</p>
```

Abbildung 4:Beispiel für eine Satzdatei, in der Sprachbeschriftungen und Vertrauens-Scores zu Satz-Tags hinzugefügt wurden.

AlignmentParser —Modul analysiert eine bestimmte XCES-Link-Datei und initialisiert SentenceParser —Module zum Parsender Sätze-Dateien.AlignmentParser gibt einzelne-Satzpaarsegmente aus, während SentenceParser einzelne Sätze von beiden Seiten der Ausrichtung ausgibt.LinksAlignmentParser kann verwendet werden, wenn nur die XCES-Links benötigt werden und das Parsen der Satzdatei übersprungen werden kann.Für das Satz-Parsing gibt es auch ein alternativesModul ExhaustiveSentenceParser,das robuster ist als SentenceParser, aber etwas langsamer, wenn nur ein kleiner Teil eines großen Korpus parsiert wird.Jedes der Module in Parse —Unterbibliotheken kann einzelnin ein Python-Skript importiert und verwendet werden, um einzelne Sätze, Satzpaareoder XCES-Links zu extrahieren.

OpusCat ist das Python-Modul, das vomopus_cat —Skript verwendet wird und beide haben die gleiche Funktionalität beim Lesen von einsprachigen Satzdateien von OPUS.OpusCat verwendet eine modifizierte Version des SentenceParser Moduls:Beim Lesen einzelner Satzdateien muss der Satz-Parsing-Prozess nicht einer in einer Ausrichtungsdatei angegebenen Reihenfolge folgen, und der SentenceParserin OpusCat gibt einfach jeden Satz in einer Datei aus.Sowohl OpusCat als auchSentenceParser können als Python-Module importiert werden, um eine detaillierte Kontrolle über das Lesen von einsprachigen Dateien zu haben.**Das OpusGet** —Modul ermöglicht das opus_get —Skript mit Corpora—Download-Funktionen.Durch den Import des Moduls in Python-Code kann man detaillierte Informationen über OPUS-Corpora innerhalb von Python-Datenstrukturen erhalten.Diese Informationen umfassen unter anderem die Anzahl der Ausrichtungspaare, die Anzahl der Dokumente, die Anzahl der Token und die Größe in Kilobyte.

Das OpusLangid Modul hat die gleiche Funktionalität wie das opus_langid Skript:hinzufügen von Sprachbeschriftungen und Sprachidentifizierungswerten für XML-Satzdateien.Darüber hinaus enthält OpusLangid LanguageIdAdder Klasse, die verwendet werden kann, umSprachbeschriftungen zu erhalten und Identifikationsvertrauenspunkte sowohl von pycld2 als auch langid.py für einen einfachen Textsatz mit einem einzigen Funktionsaufruf zu erhalten.

3.3. Das OpusTools Perl Modul

Ein komplementäres Paket von OPUS-Tools wird als Perl-Modul mit einer permissiven MIT-Lizenz bereitgestellt.5

[5]https: //github.com/Helsinki-NLP/ OpusTools—perl enthält Befehlszeilen-Tools, die speziell für die Erstellung neuer Datensätze, aber auch generell für den schnellen Zugriff auf Daten in unterschiedlichen Formaten praktisch sind.Ein Teil der Funktionalitätwird nun durch die oben beschriebenen Implementierungen in der Python-Bibliothek ersetzt, und wir werden uns hier auf die Tools konzentrieren, die zusätzliche Anwendungsfälle unterstützen.Diese Instrumente fallen hauptsächlich in die folgenden drei Kategorien:

Konvertierungswerkzeuge: Werkzeuge, die verwendet werden können, um Daten in verschiedenen Dateiformaten und Datenmarkup zu importieren und zu exportieren.Der Hauptzweck ist es, neue Datensätze in OPUS zu importieren und Dateien zu erstellen, die mit unterschiedlichen Formaten freigegeben werden.

Ausrichtungswerkzeuge: Satz- und Wortausrichtung kann auf verschiedene Weisen verwendet werden und diese Tools bieten einige praktische Operationen auf der Oberseite der ausgerichteten Bitexte.

Andere Verarbeitungswerkzeuge: Diese Kategorie umfasst Werkzeuge für Anmerkungen und Indexierung.

In der ersten Kategorie haben wir Import-Tools wie Moses2opus, tmx2opus und xml2opus.Ex Port-Skripte umfassen opus2moses, tmx2moses, opus2text und opus2multi.

xml2opus ist ein einfaches Skript, dasbeliebigen XML-Daten Satzgrenzen hinzufügt.Die Satzgrenzenerkennung erfolgt mit Hilfe der Werkzeuge, die mit dem Europarl—Parallelkorpus (Koehn, 2005) freigegeben und im Perl-Modul Lingua-::Sentence verpackt wurden.Weitere Tools auf Basis von UD-Baumbank-Klassifikatoren werden zukünftig integriert.Inline-Tags, die Markup innerhalb von Sätzen hinzufügen, werden derzeit nicht unterstützt.

Moses2opus liest ausgerichtete Klartextdateien, wie sie in der maschinellen Übersetzung verwendet werden, mit ausgerichteten Sätzen auf derselben Zeile.[5]Das Tool wandelt die Daten in einfacheseigenständiges XML für die Korpusdaten und das XCES Align-Format für die Standoff-Satzausrichtung um, wie es innerhalb von OPUS verwendet wird.Derzeit werden nur zweisprachige Eingaben unterstützt.Einfache Textdateien enthalten keine Satzgrenzen, können aber immer noch Satzausrichtungen enthalten, die nicht eins zu eins sind.Daherfügt moses2opus Satzmarkup mit-Lingua::Sentence hinzu und passt die Standoff-Satzausrichtungentsprechend an.Das Skript unterstützt auch

[5]Der Name kommt aus dem Moses-Paket, das das Format popularisierte.

das Aufteilenvon Bitexten in kleinere Teile.Leere Zeilen in der Quelle und

Zielsprache kann verwendet werden, um Dokumentgrenzen anzuzeigen.Darüber hinaus kann ein Korpus mit einer Längenschwelle für die maximale Anzahl der Übersetzungseinheiten, die in einem Teil enthalten sind, in gleich große Portionen aufgeteilt werden.

tmx2opus wandelt Übersetzungsspeicher im TMX-Format in OPUS XML um.Das Tool fügt Satzgrenzen auf die gleiche Weise wie moses2opus hinzu.Es ermöglicht auch, mehrere TMX-Dateien durch das Konvertierungstool zu pipen und es ist in der Lage, Informationen im Falle von überlappenden Sätzen, die in mehreren Übersetzungseinheiten abgedeckt sind, zusammenzuführen.Dies ist praktisch bei der Verarbeitung von Daten, die als unterschiedliche Bitexte kommen, aber denselben Inhalt abdecken.Daher werden im resultierenden OPUS XML für jede Sprache nur eindeutige Sätze gespeichert, obwohl sie in verschiedenen Übersetzungseinheiten mit Ausrichtungen zu verschiedenen Sprachen erscheinen. tmx2opus kann auch Übersetzungsspeichermit mehr als zwei Sprachen in einer Übersetzungseinheit verarbeiten,und es erzeugt zweisprachige Satzausrichtungsdateien für alle Sprachpaare, wie sie in OPUS notwendig sind.Darüber hinaus ist es auch möglich, Daten in kleinere Teile zu teilen, ähnlich dem, was moses2opus tut.Eigenschaften aus TMX-Dateien können auch in die konvertierten Daten kopiert werden, um zusätzliche Metadaten zu speichern.Die Anwendung von tmx2opus für die Schaffung des importierten ParaCrawl Corpusin OPUS ist in Abschnitt 4 beschrieben.
Exportieren Sie Skripte hauptsächlich Datenkonvertierung in die entgegengesetzte Richtung. opus2mosesund opus2textkonvertieren OPUS XML-Daten in Klartext und sie sind meist veraltet und ersetzt durch die Implementierung des zuvor eingeführten Python-Pakets. tmx2mosesist ein praktisches Skript, um aus beliebigen TMX-Dateien ausgerichtete Sätze zu extrahieren und ist nicht auf OPUS-Daten beschränkt.

opus2multi ist ein Tool, das multiparallele Datensätze aus OPUS corpora erstellen kann.In OPUS sind alle Datensätze zweisprachig ausgerichtet, aber in einigen Fällen möchte man eine Ausrichtung haben, die mehr als zwei Sprachen umfasst.Dazukann opus2multi helfen, zweisprachige Satzausrichten zu verbindenund Links über eine größere Anzahl von Sprachen zu extrahieren.Das Tool arbeitet auf Standoff-Satzausrichtungsdateien und nutzt eine Pivotsprache, um Übersetzungseinheitenüber alle angegebenen Sprachen hinweg zu konstruieren.Dazu erweitert sieteilweise überlappende Satzausrichtungen, bis alle Sprachen ohne weitere Konflikte in der resultierenden Übersetzungseinheit abgedeckt sind (d. h. keine verbleibenden Überschneidungen mit anderen Einheiten).Das Ergebnis dieses Prozessessind Satzausrichtungsdateien, die (zum Zweck der Bequemlichkeit) zweisprachig mit dem XCES Align-Format gedruckt werden, die dann mit OpusTools weiterverarbeitet werden können, um die eigentlichen Ausrichtungspaare zu extrahieren.Es besteht auch die Möglichkeit, die maximale Größe einer Übersetzungseinheit (in der Anzahl der Sätze in einer Sprache) zu kontrollieren, da die Größe im Erweiterungsprozess ohne Grenzen wachsen kann.Ein experimentelles Merkmal der Einbeziehung intralingualer Links für weitere transitive Kartierungen ist ebenfalls enthalten.Dies ist praktisch für Datensätze wie OpenSubtitles, in denen alternativeUntertiteldateien für die Verknüpfung zwischen verschiedenen Sprachen verwendet werden können.

Aligning-Tools im OpusTools-Paket helfen,Satz-Alignungen in ihrem Standoff-Annotationsformat zu verarbeiten. opus—swap-align tauscht einfach die Ausrichtungsrichtung aus.Opus bietet nur Ausrichtungen in eine Richtung

(wie sie ohnehin symmetrisch sind), aber manchmal ist es auch bequem, Zugriff auf die Links in die andere Richtung zu haben.opus —merge-align kombiniert Satzausrichtungsdateienundlöscht Duplikate, falls vorhanden. opus-split-align teilt Satzausrichtungsdateien inseparate Dateien mit einer pro Ausrichtungsgruppe, d. h. ausgerichtetes Dokument, auf.Schließlichermöglicht opus— pivoting eine transitive Satzausrichtung zwischen zwei Sprachenmittels einer Pivot-Sprache und Links zur Pivot-Sprache.Dies ist praktisch für Korpora, die mit Bitexten kommen, die nicht alle Sprachpaare abdecken, sondern nur auf eine bestimmte Sprache wie Englisch ausgerichtet sind.Unter der Annahme, dass es erhebliche Überschneidungen zwischen den Bitexten gibt, lassen Sie uns A ^ P und B ^ P sagen, Opus-pivoting extrahiert Verbindungen zwischen Sätzenin A und B und erzeugt einen neuen Bitext A ^ B. Abschnitt 4.veranschaulicht die Verwendung am Beispiel der Schaffung von MultiParaCrawl.Schließlich extrahiert ein weiteres Aligning-Tool, opus—pt2dice,grobe probabilistische zweisprachige Wörterbücher aus Phrase-Übersetzungs-Tabellen, die aus der Moses-Toolbox mit Hilfe von SMT-Tools erstellt wurden.Diese Wörterbücher verwenden einige Heuristiken, um die Daten zu filtern und das Tool erstellt auch zusätzliche Dice-Scores als symmetrisierten Ausrichtungswert aus den bedingten Übersetzungswahrscheinlichkeiten, die in den ursprünglichen Phrasentabellenenthalten sind, was für die zweisprachige Lexikonextraktion nützlich ist (Smadja et al., 1996).

Andere Werkzeuge: Die letzte Werkzeugkategorie enthält zusätzlicheDatenverarbeitungswerkzeuge wie opus —udpipe und opus —index.Erstere implementiert einen Wrapper um UDPipe (Straka und Strakova, 2017) um OPUS-Daten zu notieren und das Ergebnis in OPUS-konformem XML zu speichern.OpusTools kann vortrainierte Modelle von LINDAT verwenden.[6]Nicht zuletztist Opus-Index ein Instrument zur Indexierung von OPUS-Corpora mit der Corpus Work Bench (CWB) (Evert and Hardie, 2011).Es erstellt alle Importdateien und führt den Encoder aus, wenn verfügbar, um Multiparallelkorpuszu erstellen, die mit der CWB-Suchmaschine abgefragt werden sollen.

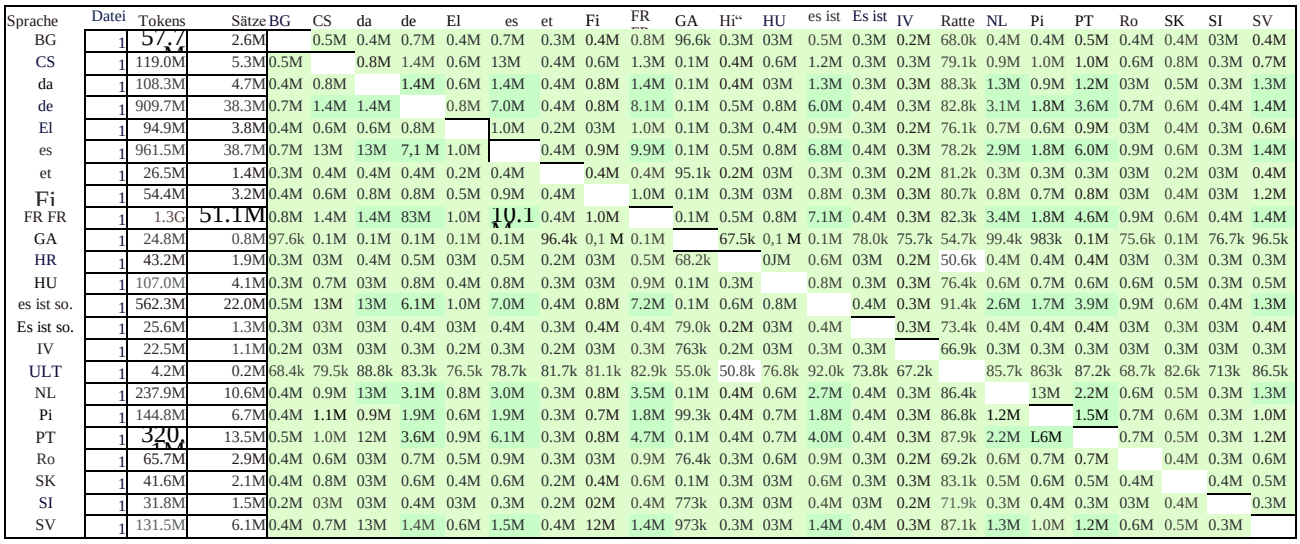
4. ParaCrawl und MultiParaCrawl

In diesem Abschnitt möchten wir den Import der ParaCrawl-Daten vorstellen, um die Verwendung von OpusTools zu demonstrieren.Der ParaCrawl-Corpus[7] wurdeextrahiert, indem man das Web krabbelte und eine komplexe Dokument- und Satzausrichtungspipelineauf Basis des Bitextor-Pakets anwendete (Espla-Gomis, 2009).Die aktuelle Version v5.0 umfasst 24 europäische Sprachenund das Projekt bietet automatisch gereinigte Bitexte fürSprachen, die auf Englisch ausgerichtet sind.Die Größe reicht von 100.000 Übersetzungseinheiten (Maltesisch-Englisch) bis zu über 50 MillionenEinheiten (Französisch-Englisch) und die Dateien werden in Klartext oder TMX-Format verteilt.Während es ein paar Bonus-Sprachpaare gibt, die auch zwei Bitexte enthalten, die nicht Englisch enthalten, ist die Mehrheit der Sammlung zweisprachig mit englischen Inhalten ausgerichtet.

Ziel der Integration von ParaCrawl in OPUS ist es, die Daten über das native OPUS-Format zur Verfügung zu stellen und auch alle in der Sammlung enthaltenen Sprachpaare umfassend abzudecken.Zu diesen Zwecken hat die zuvor eingeführte

[6]https://lindat.mff.cuni.cz

[7]https://paracrawl.eu



Sprache	Datei	Tokens	Sätze	BG	CS	da	de	El	es	et	Fi	FR	GA	Hi“	HU	es ist	Es ist	IV	Ratte	NL	Pi	PT	Ro	SK	SI	SV
BG	1	57.7	2.6M		0.5M	0.4M	0.7M	0.4M	0.7M	0.3M	0.4M	0.8M	96.6k	0.3M	03M	0.5M	0.3M	0.2M	68.0k	0.4M	0.4M	0.5M	0.4M	0.4M	03M	0.4M
CS	1	119.0M	5.3M	0.5M		0.8M	1.4M	0.6M	13M	0.4M	0.6M	1.3M	0.1M	0.4M	0.6M	1.2M	0.3M	0.3M	79.1k	0.9M	1.0M	1.0M	0.6M	0.8M	0.3M	0.7M
da	1	108.3M	4.7M	0.4M	0.8M		1.4M	0.6M	1.4M	0.4M	0.8M	1.4M	0.1M	0.4M	03M	1.3M	0.3M	0.3M	88.3k	1.3M	0.9M	1.2M	03M	0.5M	0.3M	1.3M
de	1	909.7M	38.3M	0.7M	1.4M	1.4M		0.8M	7.0M	0.4M	0.8M	8.1M	0.1M	0.5M	0.8M	6.0M	0.4M	0.3M	82.8k	3.1M	1.8M	3.6M	0.7M	0.6M	0.4M	1.4M
El	1	94.9M	3.8M	0.4M	0.6M	0.6M	0.8M		1.0M	0.2M	03M	1.0M	0.1M	0.3M	0.4M	0.9M	0.3M	0.2M	76.1k	0.7M	0.6M	0.9M	03M	0.4M	0.3M	0.6M
es	1	961.5M	38.7M	0.7M	13M	13M	7,1 M	1.0M		0.4M	0.9M	9.9M	0.1M	0.5M	0.8M	6.8M	0.4M	0.3M	78.2k	2.9M	1.8M	6.0M	0.9M	0.6M	0.3M	1.4M
et	1	26.5M	1.4M	0.3M	0.4M	0.4M	0.4M	0.2M	0.4M		0.4M	0.4M	95.1k	0.2M	03M	0.3M	0.3M	0.2M	81.2k	0.3M	0.3M	0.3M	03M	0.2M	03M	0.4M
Fi	1	54.4M	3.2M	0.4M	0.6M	0.8M	0.8M	0.5M	0.9M	0.4M		1.0M	0.1M	0.3M	03M	0.8M	0.3M	0.3M	80.7k	0.8M	0.7M	0.8M	03M	0.4M	03M	1.2M
FR FR	1	1.3G	51.1M	0.8M	1.4M	1.4M	83M	1.0M	10.1	0.4M	1.0M		0.1M	0.5M	0.8M	7.1M	0.4M	0.3M	82.3k	3.4M	1.8M	4.6M	0.9M	0.6M	0.4M	1.4M
GA	1	24.8M	0.8M	97.6k	0.1M	0.1M	0.1M	0.1M	0.1M	96.4k	0,1 M	0.1M		67.5k	0,1 M	0.1M	78.0k	75.7k	54.7k	99.4k	983k	0.1M	75.6k	0.1M	76.7k	96.5k
HR	1	43.2M	1.9M	0.3M	03M	0.4M	0.5M	03M	0.5M	0.2M	03M	0.5M	68.2k		0JM	0.6M	03M	0.2M	50.6k	0.4M	0.4M	0.4M	03M	0.3M	0.3M	0.3M
HU	1	107.0M	4.1M	0.3M	0.7M	03M	0.8M	0.4M	0.8M	0.3M	03M	0.9M	0.1M	0.3M		0.8M	0.3M	0.3M	76.4k	0.6M	0.7M	0.6M	0.6M	0.5M	0.3M	0.5M
es ist so.	1	562.3M	22.0M	0.5M	13M	13M	6.1M	1.0M	7.0M	0.4M	0.8M	7.2M	0.1M	0.6M	0.8M		0.4M	0.3M	91.4k	2.6M	1.7M	3.9M	0.9M	0.6M	0.4M	1.3M
Es ist so.	1	25.6M	1.3M	0.3M	03M	03M	0.4M	03M	0.4M	0.3M	0.4M	0.4M	79.0k	0.2M	03M	0.4M		0.3M	73.4k	0.4M	0.4M	0.4M	03M	0.3M	03M	0.4M
IV	1	22.5M	1.1M	0.2M	03M	03M	0.3M	0.2M	0.3M	0.2M	03M	0.3M	763k	0.2M	03M	0.3M	0.3M		66.9k	0.3M	0.3M	0.3M	03M	0.3M	03M	0.3M
ULT	1	4.2M	0.2M	68.4k	79.5k	88.8k	83.3k	76.5k	78.7k	81.7k	81.1k	82.9k	55.0k	50.8k	76.8k	92.0k	73.8k	67.2k		85.7k	863k	87.2k	68.7k	82.6k	713k	86.5k
NL	1	237.9M	10.6M	0.4M	0.9M	13M	3.1M	0.8M	3.0M	0.3M	0.8M	3.5M	0.1M	0.4M	0.6M	2.7M	0.4M	0.3M	86.4k		13M	2.2M	0.6M	0.5M	0.3M	1.3M
Pi	1	144.8M	6.7M	0.4M	1.1M	0.9M	1.9M	0.6M	1.9M	0.3M	0.7M	1.8M	99.3k	0.4M	0.7M	1.8M	0.4M	0.3M	86.8k	1.2M		1.5M	0.7M	0.6M	0.3M	1.0M
PT	1	320.	13.5M	0.5M	1.0M	12M	3.6M	0.9M	6.1M	0.3M	0.8M	4.7M	0.1M	0.4M	0.7M	4.0M	0.4M	0.3M	87.9k	2.2M	L6M		0.7M	0.5M	0.3M	1.2M
Ro	1	65.7M	2.9M	0.4M	0.6M	03M	0.7M	0.5M	0.9M	0.3M	03M	0.9M	76.4k	0.3M	0.6M	0.9M	0.3M	0.2M	69.2k	0.6M	0.7M	0.7M		0.4M	0.3M	0.6M
SK	1	41.6M	2.1M	0.4M	0.8M	03M	0.6M	0.4M	0.6M	0.2M	0.4M	0.6M	0.1M	0.3M	03M	0.6M	0.3M	0.3M	83.1k	0.5M	0.6M	0.5M	0.4M		0.4M	0.5M
SI	1	31.8M	1.5M	0.2M	03M	03M	0.4M	03M	0.3M	0.2M	02M	0.4M	773k	0.3M	03M	0.4M	03M	0.2M	71.9k	0.3M	0.4M	0.3M	03M	0.4M		0.3M
SV	1	131.5M	6.1M	0.4M	0.7M	13M	1.4M	0.6M	1.5M	0.4M	12M	1.4M	973k	0.3M	03M	1.4M	0.4M	0.3M	87.1k	1.3M	1.0M	1.2M	0.6M	0.5M	0.3M	

Abbildung 5:Statistiken aus dem MultiParaCrawl Korpus – eine mehrsprachige Erweiterung von ParaCrawl über Pivot Ausrichtung durch Englisch.Das obere rechte Dreieck gibt die Größe in Bezug auf Satzausrichtung im Klartextformat an, und das untere linke Dreieck zeigt die Größe der extrahierten TMX-Dateien in Bezug auf einzigartige Übersetzungseinheiten pro Sprachpaar an.

Werkzeuge tmx2opus und opus-pivoting werden praktisch. tmx2opus ist nicht nur nützlich, um die Ausrichtungen aus der ursprünglichen TMX-Quelle zu extrahieren, sondern bietet auch die Funktionalität, um Satzgrenze Markuphinzuzufügen und Redundanz zwischen den verschiedenen Bitexten zu reduzieren.Die Verwendung der einzigartigen Option von tmx2opus reduziert die Größe des englischen Teilsdes Korpus (d. h. 252 Millionen getrennt ausgerichtete englische Sätzein 23 Bitexten) auf weniger als 60 % der ursprünglichen Daten.-Gleichzeitig ermöglicht das Einzigartigkeitsmerkmal auch denAufbau eines multiparallelen Korpus, indem es in dem neu geschaffenen Satz von Sätzen auf Englisch schwenkt.Dazukann Opus-Pivoting wie bereits erläutert verwendet werden.Mit diesem Verfahren konnten 253 zusätzliche Bitextemit Größen von bis zu 10 Millionen satzgebundenen Übersetzungseinheiten erstellt werden.Abbildung 5 fasst die nicht-englischen Bitexte in MultiParaCrawl zusammen.

5. Parallele Corpus-Diagnose

Unsere Diagnoseroutine für die OPUS-Sammlung verwendet das Kommandozeilenprogramm opus_read (beschrieben in Abschnitt3.1.), um ausgerichtete Klartextdaten für ein bestimmtes Sprachpaar ineinem bestimmten Korpus abzurufen.Um dies zu tun,analysiert opus_readdie nativen XML-formatierten Daten, um die angeforderteTeilmenge von Daten zu generieren, und führt dann eine Konvertierung in das reineTextformat durch.Während dieses Prozesses lauscht die-Diagnoseroutine auf eventuell auftretende Fehler und protokolliert sie, um einen Diagnosebericht für eine spätere Analyse zu erstellen.Wir führen dieses Verfahren systematisch für alle Sprachen, die unter jedem OPUS-Corpora verfügbar sind, durch.[8]Für unsere Diagnostik nutzen wir die volle Granularität von OPUS, indem wir getrennte Messwerte für verschiedene Korpora sammeln, die Bitexte zusammenstellen, und auch regionale Varianten vonSprachen getrennt halten, anstatt sie zu verdichten.Um diese Art von erschöpfenden Analysen durchzuführen, haben wir insgesamt 87.948 CPU-Array-Jobs parallel ausgeführt, wobei Laufzeiten zwischen

[8]Wir haben keine Diagnostik über die beiden jüngsten Ergänzungenzu OPUS durchgeführt:Infopankki und MultiParaCrawl.

1 Sekunde und 5,2 Stunden variierten und jeder Job zwischen 4und 128 GB Arbeitsspeicher verwendete.Insgesamt dauerte die gesamte diagnostischeAnalyse etwa 1000 Stunden Berechnung,durchschnittlich 18,2 Stunden pro Korpus.Während die Granularitätunserer Analyse intern nützlich sein wird, um Anomalien in OPUS zu identifizieren, um Reparaturen zu erleichtern, sammeln wir auch unsere Daten, um korpusweite Zahlen zu generieren, die wir in diesem Abschnitt berichten und diskutieren.

5.1. Fehleranalyse

Die in unserem Bericht angemeldeten „Diagnosen“ listen die Ursachen jedes Abruffehlers auf, wodurch wir sie zuverlässig lokalisieren und beheben können.Bei der Zusammenstellung aller Diagnosen zeigen die Ergebnisse, dass zwar 37 der Korpora vollständig fehlerfrei sind, die Datenabfrage jedoch für mindestens ein Sprachpaar für die restlichen 18 Korpora gestoppt wurde.Die Häufigkeit vonAbruffehlern in diesen Korpus variiert von einem winzigen Bruchteil zum gesamten Korpus (siehe Abbildung 8).Die überwiegende Mehrheitdieser Fehler resultiert aus unzulässigen XML-Daten mit ungültigen Token (96,2 %) oder unpassenden Tags (3,5 %).Unsere bisherigen Teilprüfungen deuten darauf hin, dass diese auf geringfügige Konvertierungsfehler wie ungenutzte Sonderzeichenund XML-Entitäten zurückzuführen sind, die in den ursprünglichen Daten vor dem Import in OPUS vorkommen.Ein weiterer sehr kleiner Teil der Fehler (0,3 %) weist auf fehlende Datendateien im Hauptdateisystem hin, in dem OPUS gehostet wird, was wahrscheinlich Kopierfehler anzeigt und noch weiter untersucht werden muss.

5.2. Korpus-Wide Statistik

Neben der Katalogisierung der Datenabfrage berechnet unser diagnostisches Verfahren auch einige grundlegende quantitativeStatistiken, wie z. B. die gemeldeten Rechenkosten für die Datenabfrage und verschiedene Messungen der proKorpus, Sprache und Sprachpaar abgerufenen Daten.Unsere entsprechendenstatistischen Analysen ergaben zum größten Teil keine nennenswerten Trends oder Ausreißer, mit Ausnahme einiger Messungen, dieauf die relativen Abweichungen und den Lärmpegel in Datenüber Korpora hindeuteten.In den Abbildungen 6 und 7 berichten wir

Verteilung von zwei Maßen über die verfügbare durchschnittliche Satzlänge in Zeichen.Beide Guage-Paare für jeden Korpus:die durchschnittliche Anzahl der sen-Maße wurde anhand von Box-und-Whisker-Plots zu Tence-Paaren (oder genauer: Übersetzungseinheiten) visualisiert und die Verteilungsunterschiede betont, wobei die Endpunkte

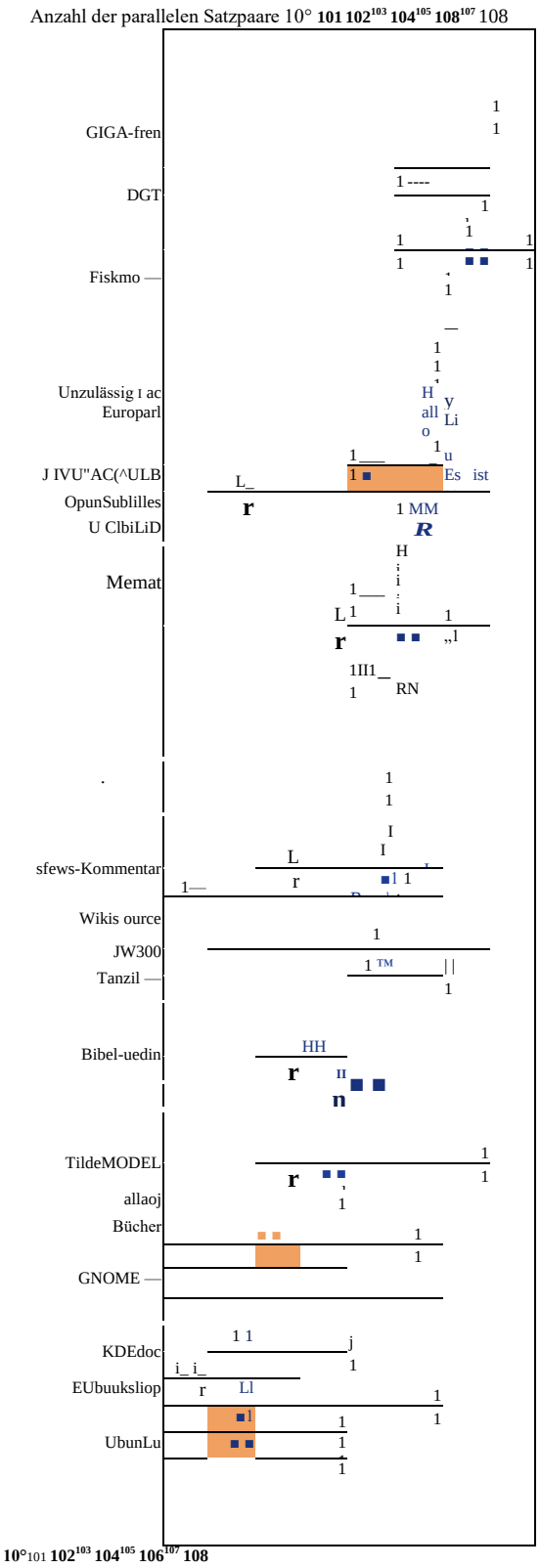


Abbildung 6:Verteilung der Anzahl der abrufbaren parallelen Satzpaare über den Satz der verfügbaren Sprachpaare für jeden Korpus.

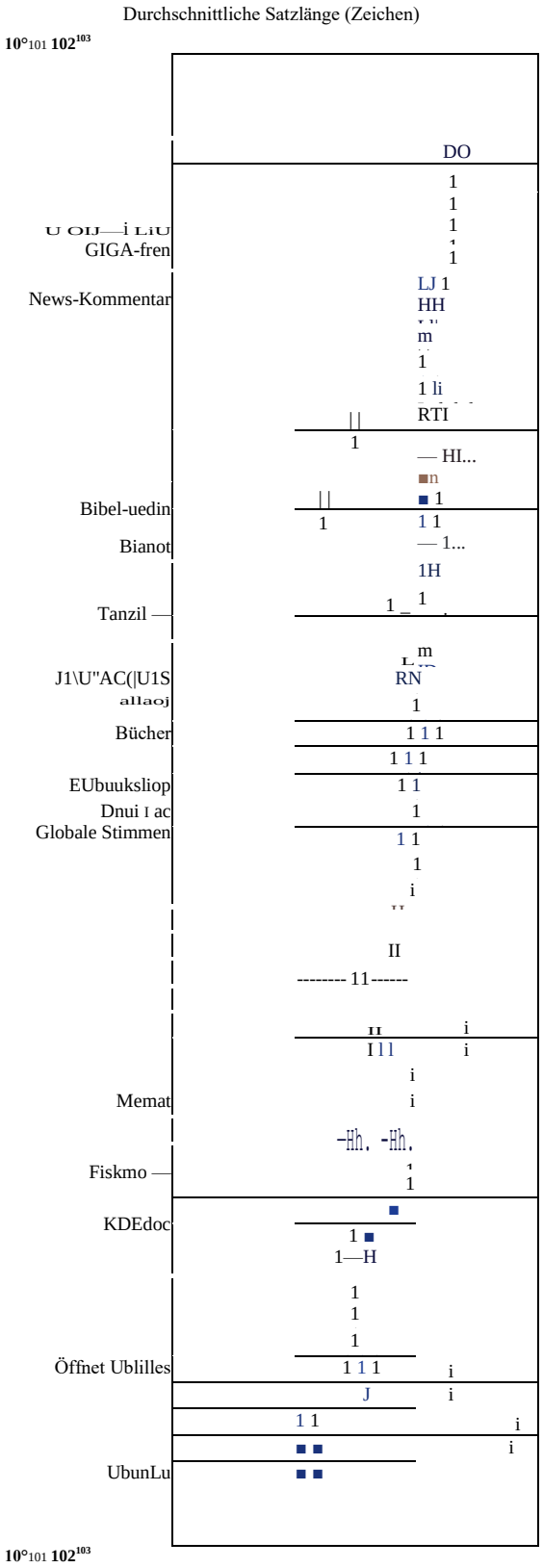


Abbildung 7:Verteilung der durchschnittlichen Satzlängen (in Zeichen) über Sätze verfügbarer Sprachpaare für jeden

Korpus.

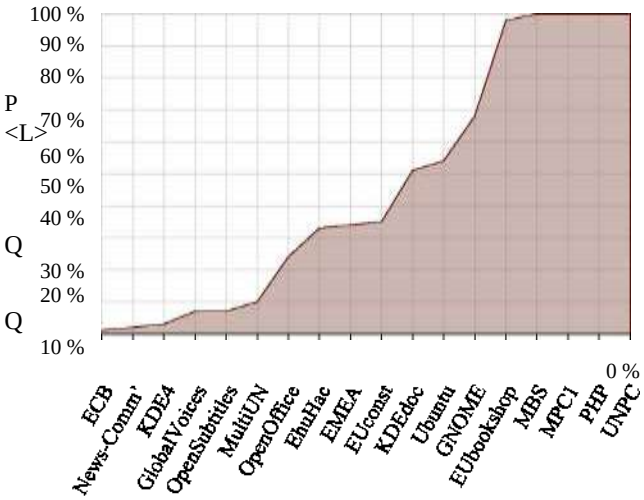


Abbildung 8:Prozentsätze von Sprachpaaren in OPUS-Corpora, bei denen die Datenabfrage mit OPUS-Tools Fehler zurückgibt.Fehlerfreie Korpus wurden aus dem Graphen weggelassen.

Zeigen Sie die niedrigsten und höchsten Werte,[9]und die beiden Hälften der Box stellen das zweite und dritte Quartil von Werten, getrennt durch den Median.

Eines der markantesten Details aus Abbildung 6 ist der Kontrastzwischen den Abweichungen.Mehr als ein Drittel der Korpora zeigen sehr wenig bis keine Varianz zwischen Sprachpaaren, was völlig multiparallele Daten impliziert, während andere wie JW300 und OpenSubtitles umgekehrt sehr hohe Varianz aufweisen, wobei der Unterschied in den Größen der verfügbaren Daten mehrere Größenordnungen umfassen kann.Bei den ersten Quartilen scheinen einige Korpora wie QED und Tatoeba einen signifikanten Anteil an Sprachpaaren zu haben, die nursehr wenige Übersetzungseinheiten enthalten, was möglicherweise auf hohe Spracherkennungoder Satzausrichtungsgeräusche hindeutet.In Abbildung 7 scheint das erste Quartil für einige Korpora einen ähnlichen relativen Bereich zu haben, was bedeutet, dass Sätze durchschnittlich nur wenige Zeichen für einige der verfügbaren Sprachpaare enthalten.Es ist wahrscheinlich kein Zufall, dass diese Fälle meist Korpora entsprechen, die aus natürlich lauten Daten zusammengestellt wurden.Darüber hinaus deuten die kleinsten und größten Medianwerte in Abbildung 7 auf außergewöhnlich kurze und außergewöhnlichlange „typische“ Sätze im entsprechenden Korpus hin,die auf einen starken Kontrast in der Textsegmentierung oder aufdeutlich unterschiedliche Datendomänen hinweisen können.Die drei Korpora mit den niedrigsten Medianen umfassen zum Beispiel Übersetzungen-von Computersoftware, während Dokumente aus den Vereinten Nationen die höchste Medianlänge aufweisen.

6. Schlussfolgerungen und künftige Arbeiten

In diesem Beitrag stellen wir OpusTools vor, ein Open-Source-Paket aus Bibliotheken und Kommandozeilen-Tools für einen effizienten und komfortablen Zugriff auf parallele Korpora in der umfangreichen OPUS-Datensammlung.Das Paket implementiert Werkzeuge zum Herunterladen, Konvertieren, Filtern und Verarbeiten parallelerDatensätze und erleichtert den Zugriff auf komprimierte und archivierte Dateien aus der Sammlung.Es bietet auch eine Python-Bibliothek für den programmatischen Zugriff auf die Daten, so dass es einfach ist, Datenverarbeitung in die Entwicklung anderer Tools zu integrieren

[9]Ungedeckte Endpunkte zeigen Extrema über die Grenzen der x-Achse hinaus an.

.Darüber hinaus stellen wir Werkzeuge für die Datenkonvertierung und -ausrichtung vor, die bei der Erstellung neuer Datensätze aus verschiedenen Quellen angewendet werden können.Wir zeigen ihre Verwendung am Beispiel des kürzlich hinzugefügten MultiParaCrawl Corpus, der den ursprünglichen Datensatz um pivot-basierte Ausrichtungenzwischen allen Sprachpaaren erweitert, die zur wachsenden Abdeckungder OPUS-Datenbank beitragen.

Obwohl es ziemlich schwierig ist, eine Kollektion so groß wie OPUS zu halten, wird die Fehlerbehebung einfacher und schneller sein, wenn die Diagnose vollständig aufgezeichnet ist.Insgesamt scheinen die statistischen Analysen eher auf eine bemerkenswerte qualitativeund quantitative Diversität zwischen OPUS Korpora hindeuten, wobei Trends scheinbar innerhalb der Erwartungen liegen, und Randfälle, die dem Rauschen in den ursprünglichen Daten zugeschrieben werden können.Unsere Absicht ist es, alle Probleme rund um das Abrufen von Daten zu lösen, so dass die Verwendung von OPUS-Tools eine reibungslose Erfahrung für alle Benutzer sein wird, und auch unsere Routine als Diagnose-Tool zu straffen, das zu einem Standard-Teil des Prozesses der Erweiterung von OPUS um neue Korpora werden würde.

Danksagungen

ERC Diese Arbeit ist Teil des FoTran-Projekts, das vom Europäischen Forschungsrat (ERC) finanziert wird. Das Forschungs—und Innovationsprogramm „Horizont 2020“ der Europäischen Union (Zuschussvereinbarung Na 771113)sowie das MeMAD-Projekt, das aus dem Forschungs- und Innovationsprogramm „Horizont 2020“ der Europäischen Union (-Zuschussvereinbarung Na 780069) finanziert wird.

7. Bibliographische Referenzen

Evert, S. und Hardie, A. (2011).. Jahrhundert Korpus-Werkbank:Aktualisierung einer Abfragearchitektur für das neue Jahrtausend.In *Proceedings of the Corpus Linguistics-2011 Conference, University of* Birmingham, UK.Smadja, F., McKeown, K. R., und Hatzivassiloglou, V. (1996).Übersetzen von Kollokationen für zweisprachige Lexikon:Ein statistischer Ansatz.*Computational Linguistics*, 22(1):1-38.

8. Referenzen zur Sprachressource

Espla-Gomis, Miquel.(2009).*Bitextor:eine kostenlose/offene Software, um Übersetzungserinnerungen von mehrsprachigen Websites zu ernten.*

Koehn, Philipp.(2005).*Europarl:Ein Parallelkorpus für statistische maschinelle Übersetzung*.AAMT. AAMT.

Lui, Marco und Baldwin, Timothy.(2012). *langid.py:Ein Off-the-shelf Language Identification Tool.*Vereinigungfür Computational Linguistics.

Straka, Mailand und Strakova, Jana.(2017).*Tokenizing, POS Tagging, Lemmatizing und Parsing UD 2.0 mit UDPipe.*Vereinigung für Computational Linguistics.

Tiedemann, Jorg.(2012).*Parallele Daten, Tools und Schnittstellenin OPUS.*European Language Resources Association(ELRA).