LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Tokenisierungsansätze

Eine Gegenüberstellung von Tokenisierungsansätzen für große Sprachmodelle

Ioannis Partalas
Profilierungsmodul I - CIS LMU
Dr. Robert Zangenfeind
23.01.2024

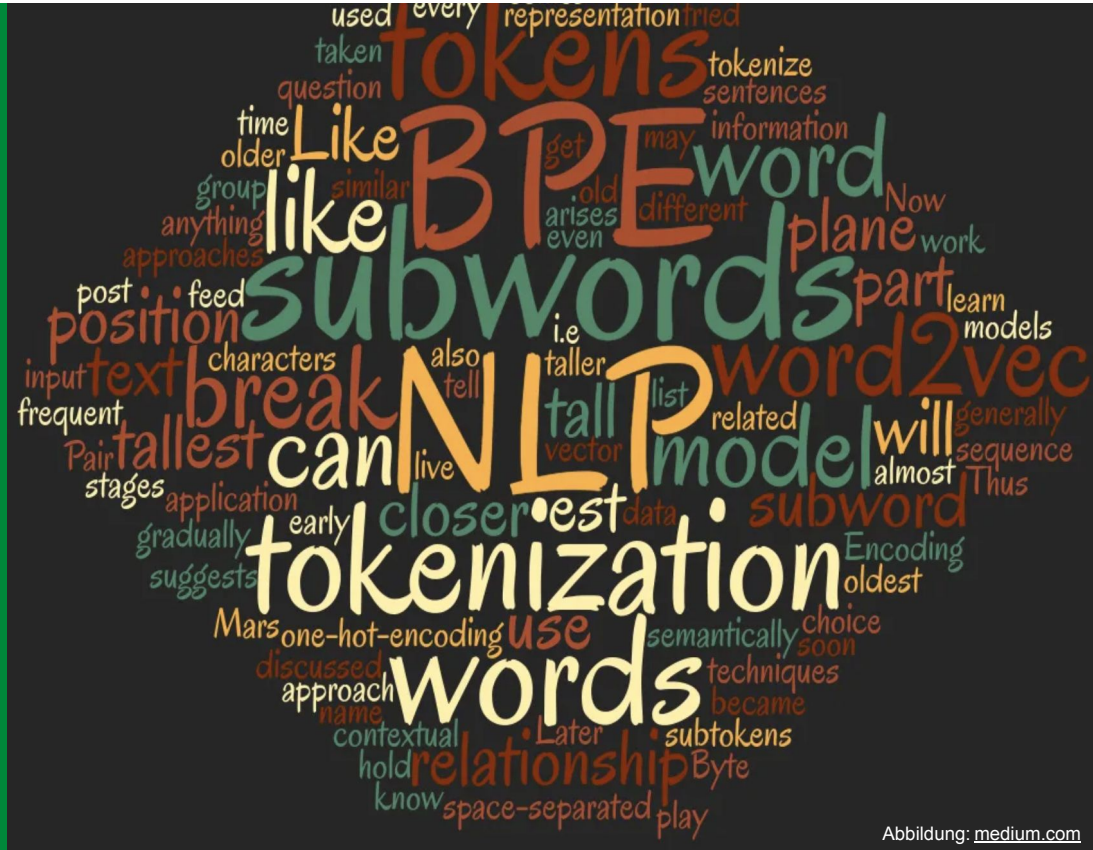


Abbildung: medium.com

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Inhaltsverzeichnis

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Einleitung
NLP Pipeline

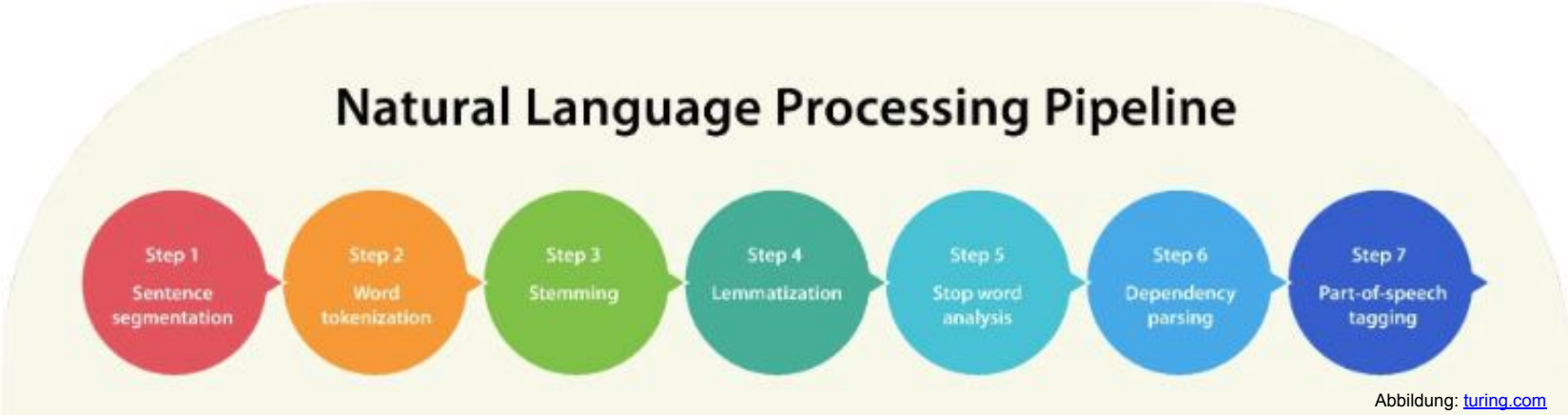


Abbildung: turing.com

- NLP-Projekt → Textvorverarbeitung → Daten in eine analysierbare Form zu bringen
  - Wichtigster Schritt → Tokenisierung
  - Zerlegung von Textdaten in kleinere Teile, die als Token bezeichnet werden
    - Wörter
    - Phrasen
    - Zeichen
    - Teilwörter

Einleitung
Tokenisierungstechniken

- Jeder Ansatz hat Vor- und Nachteile
- Wahl des Ansatzes
  - abhängig von den spezifischen Anforderungen der Aufgabe
  - beeinflusst die Genauigkeit und Effizienz von “downstream” NLP Aufgaben

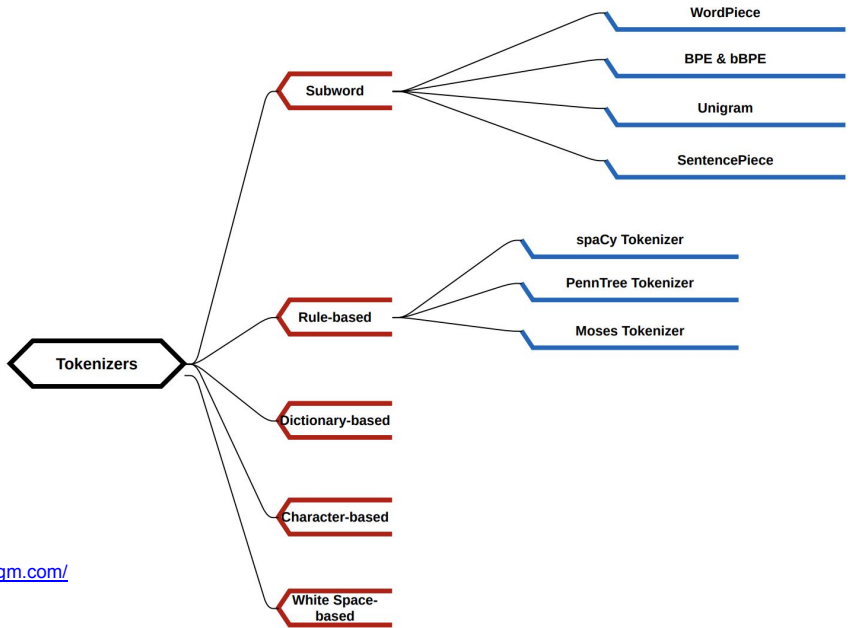


Abbildung: Erstellt mit online.visual-paradigm.com/

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Statistisch motivierte Tokenisierungsansätze

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Statistisch motivierte Subwort-Tokenisierer

Motivation[1],[2]

Vorteile gegenüber anderen Tokenisierungsansätzen → **Effizienz, Ausdruckskraft und Anpassungsfähigkeit**

- **Verwaltung der Vokabulargröße:**
  - Verwaltet effektiv die Größe des Vokabulars
  - Ausgleich zwischen zu vielen seltenen Wörtern (bei der Tokenisierung auf Wortebene) und zu vielen Token pro Satz (bei der Tokenisierung auf Zeichenebene)

- **Behandlung von Wörtern außerhalb des Vokabulars:**
  - effizient behandelt, zerlegt in kleinere bekannte Teilwörter

- **Bessere Repräsentation der Morphologie:**
  - Teilwörter können die Wortmorphologie besser erfassen
    - z.B. Un - zufrieden - heit → Präfix, Stamm, Suffix
  - Besonders vorteilhaft für morphologisch reiche Sprachen (z.B. agglutinierende Sprachen)

- **Trainingseffizienz:**
  - Führt oft zu einem effizienteren Training von LLMs
  - Bietet guten Kompromiss zwischen der Anzahl der zu verarbeitenden Token (kürzer als auf Zeichenebene) und der Aussagekraft jedes Tokens (nuancierter als auf Wortebene)

- **Agnostizismus der Sprache:** Sprachunabhängig

Statistisch motivierte Subwort-Tokenisierer[3]

- Trainierbare Tokenisierungsalgorithmen auf dem Korpus, wo das entsprechende Modell trainiert wird.
- Zwei Schritte vor Tokenisierung: Normalisierung und Pre-Tokenisierung.
- Entstehende Wörter bilden die Grenzen der Subtokens, die während des Trainings gelernt werden können.

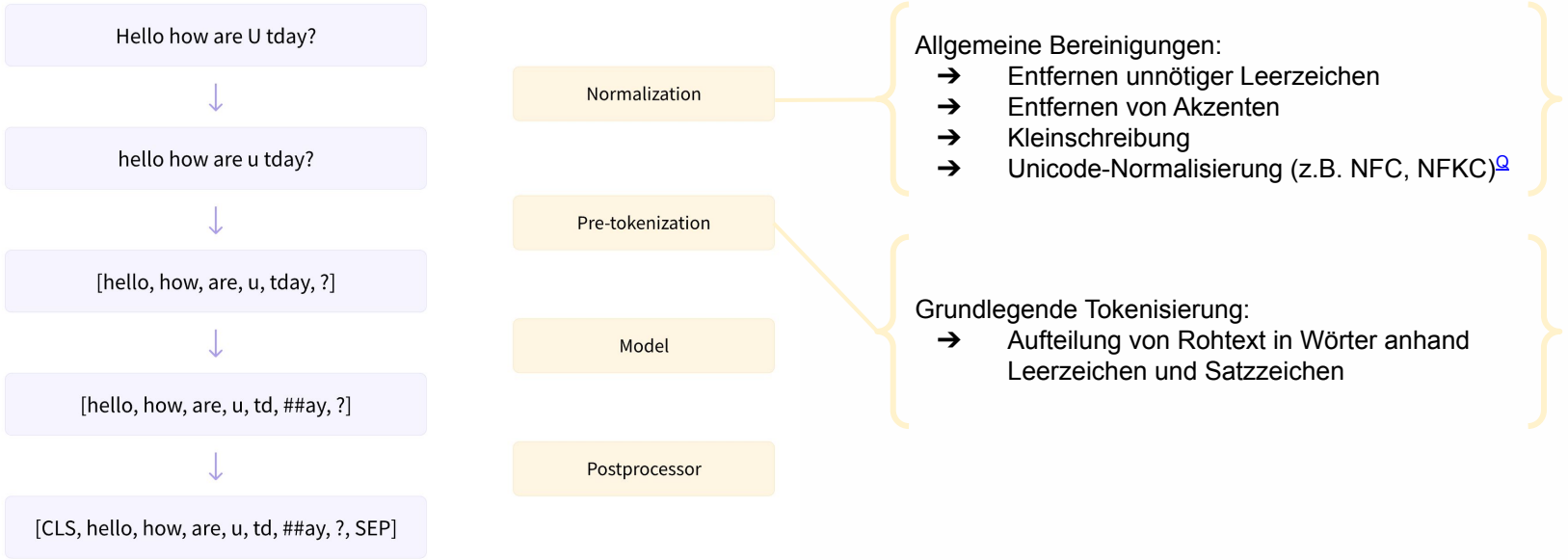


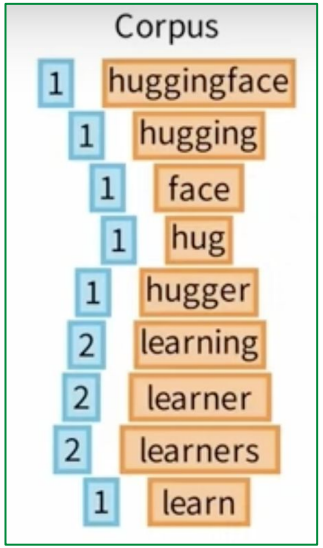
Abbildung: Hugging Face: Normalization and Pre-tokenization.

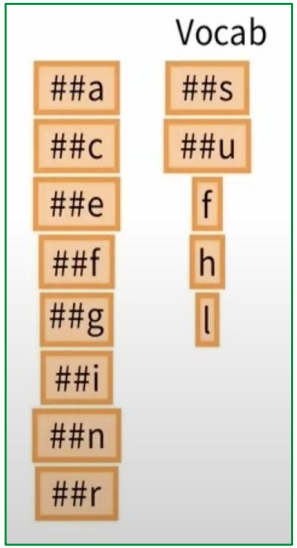
WordPiece[4],[5]

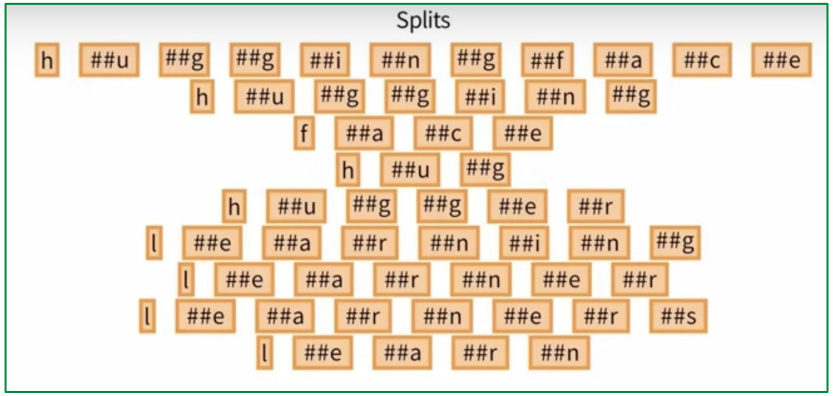
1. Wahl von Trainingskorpus und Vokabular-Größe

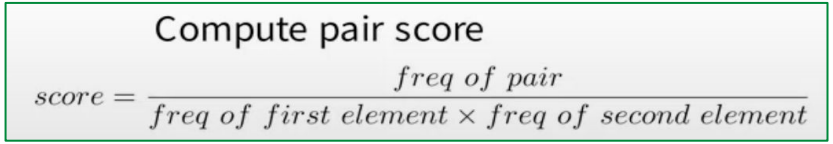
2. Aufteilung jedes Wortes in Buchstaben(-paare)
3. Erweiterung des Vokabulars

4. Auflistung der vorhandenen Splittpaare
5. Bewertung und Auswahl des höchsten Splitts
6. Erweiterung des Vokabulars









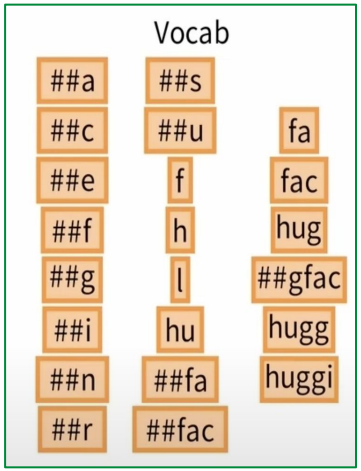
Abbildungen: Hugging Face: WordPiece tokenization

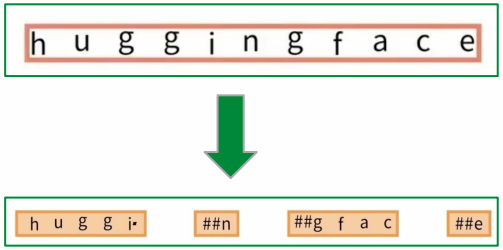
LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

WordPiece[4],[5]

7. Wiederholung der Schritte 4-6, bis Vokabular-Größe erreicht ist

Anwendung:
Am Anfang des Wortes nach dem längsten Vorkommen im Vokabular suchen, dasselbe mit den übrigen Teilstrings tun





- 2012 erfunden, 2021 optimiert
- Weit verbreitet bei *BERT* und seinen Nachfolgern: *DistilBERT*, *MobileBERT*, *FlauBERT*, *GermanBERT*, *RuBERT*, *ChineseBERT* u.a.
- Stellt jedem Teilwort, das nicht der Anfang eines Wortes ist, ein # voran.

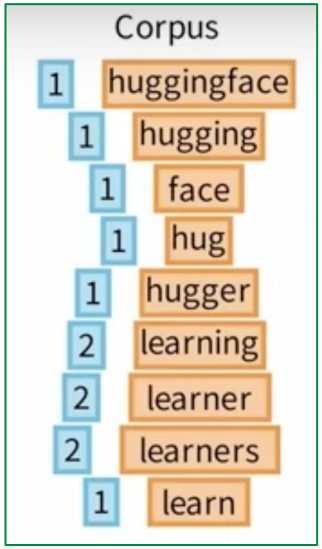
Abbildungen: Hugging Face: WordPiece tokenization

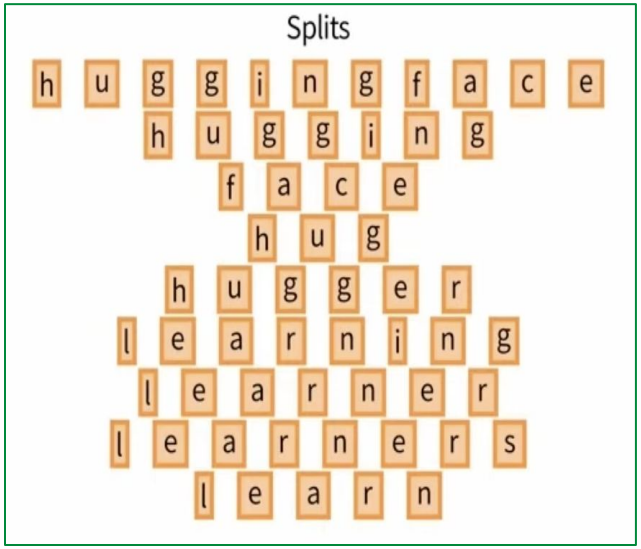
Byte-Pair Encoding[6]

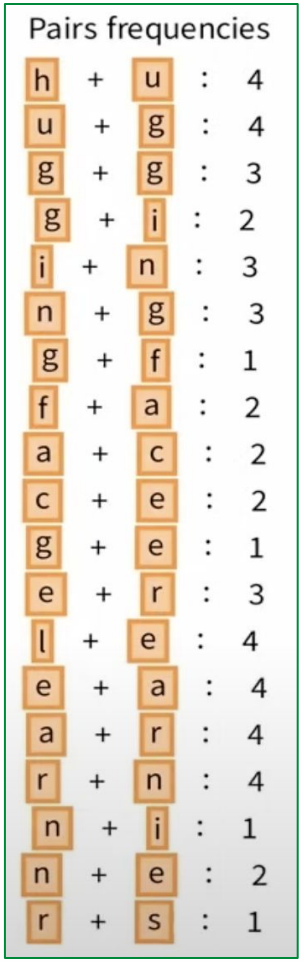
1. Wahl von Trainingskorpus und Vokabular-Größe

2. Aufteilung jedes Wortes in Buchstaben(-paare)
3. Erweiterung des Vokabulars

4. Auflistung der vorhandenen Splittpaare
5. Auszählung der Paarhäufigkeit und Auswahl des häufigsten Splittpaares
6. Erweiterung des Vokabulars





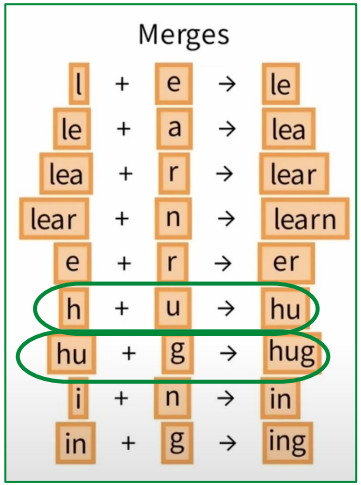


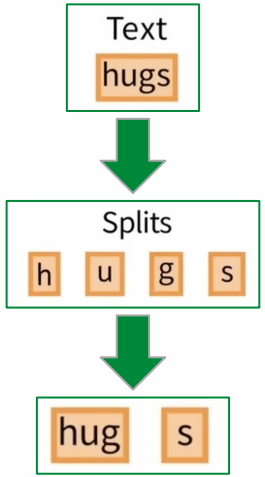
Abbildungen: Hugging Face: Byte-Pair Encoding tokenization

Byte-Pair Encoding[6]

7. Wiederholung der Schritte 4-6, bis Vokabular-Größe erreicht ist

Anwendung:
Merge-Regeln in absteigender Reihenfolge durchgehen, anwendbare Regeln anwenden, bis die letzte Regel erreicht ist



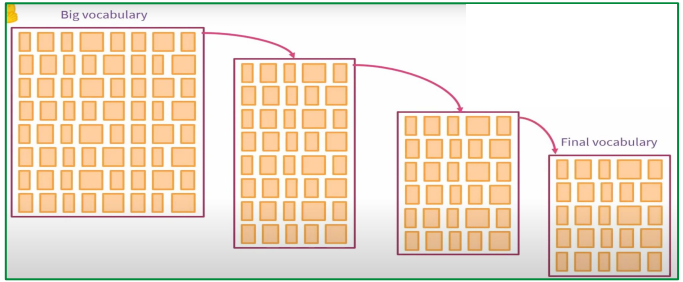


- Ursprünglich als Algorithmus zur Komprimierung von Daten entwickelt[7]
- Benutzt in *GPT*, *GPT-2*, *RoBERTa*, *BART*, *DeBERTa* u.a.
- **Byte-level Byte-Pair Encoding (bBPE)**[8], spezielle Version, die eine Sequenz auf Byte-Level konvertiert: W e s t → 57 65 73 74
  - nützlich bei der Handhabung komplexer OOV
  - effektiver als BPE bei mehrsprachigen Aufgaben, da alle existierende Zeichen kombinierbar

Abbildungen: Hugging Face: Byte-Pair Encoding tokenization

Unigramm-Tokenisierung[9]

Strategie: Großes Vokabular schrittweise zur gewünschten Vokabulargröße durch EM



1. Aufbau eines Basisvokabulars entweder mit den häufigsten Teilstrings oder mit BPE auf dem Korpus

2. Berechnung der Wahrscheinlichkeit jeder Vokabular-Einheit mit Unigramm-Modell

3. Berechnung einer Wahrscheinlichkeit für jede mögliche Wortsegmentierung und Auswahl derjenigen mit der höchsten (mit Viterbi Algorithmus)

E-Schritt: Berechnung der Wahrscheinlichkeiten

Corpus

10	hug
12	pug
5	lug
4	bug
5	dug

Vocab

h	10/180	ug	36/180
u	36/180	pu	12/180
g	36/180	hu	10/180
l	5/180	lu	5/180
p	12/180	du	5/180
b	4/180	bu	4/180
d	5/180		

Possible splits for "hug"

h u g $\frac{10}{180} \times \frac{36}{180} \times \frac{36}{180} = 2.22e-03$

hu g $\frac{10}{180} \times \frac{36}{180} = 1.11e-02$

h ug $\frac{10}{180} \times \frac{36}{180} = 1.11e-02$

hug 0

Corpus	Splits		Scores
10 hug	→ hu	g	1.11e-02
12 pug	→ pu	g	1.33e-02
5 lug	→ lu	g	5.56e-03
4 bug	→ bu	g	4.44e-03
5 dug	→ du	g	5.56e-03

Loss

$$\sum freq \times (-log(P(word)))$$

$$10 \times (-log(1.11e-02))$$
$$+12 \times (-log(1.33e-02))$$
$$+5 \times (-log(5.56e-03))$$
$$+4 \times (-log(4.44e-03))$$
$$+5 \times (-log(5.56e-03))$$

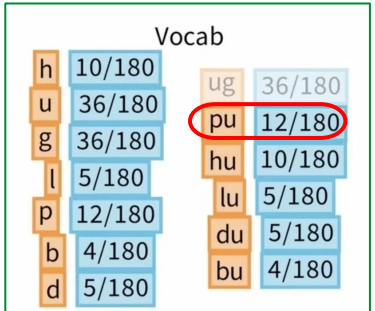
Abbildungen: Hugging Face: Unigram tokenization

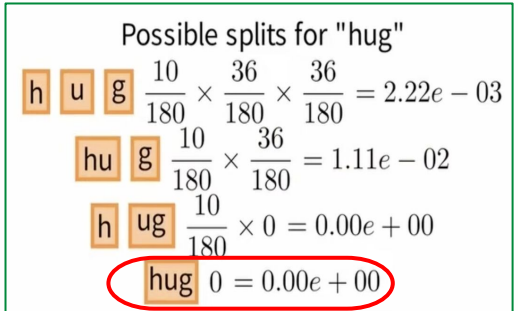
Unigramm-Tokenisierung[9]

M-Schritt: Eliminierung von Tokens, die den Verlust (loss) im Korpus am wenigsten beeinflussen

4. Eliminierung 10% der Tokens, die mit dem geringsten Zuwachs von Loss verbunden sind

5. Erneute Berechnung von Loss
6. Wiederholung der E & M-Schritte, bis Vokabular-Größe erreicht ist





- Häufig in Kombination mit dem SentencePiece Tokenisierungsalgorithmus verwendet
- Zusammen benutzt u.a. in
  - *ALBERT*,
  - *T5*,
  - *mBART*,
  - *Big Bird*,
  - *XLNet*

 Abbildungen: Hugging Face: Unigram tokenization

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

SentencePiece[10],[2]

- Problem mit bisherigen Tokenisierungsalgorithmen:
  - Annahme: Eingabetext verwendet Leerzeichen zur Trennung von Wörtern
  - Unsegmentierte Sprachen (z.B. Japanisch, Chinesisch)
- Mögliche Lösung:
  - Verwendung von sprachspezifischen Pre-Tokenizern, z. B. in XLM für Japanisch, Chinesisch und Thai
- **SentencePiece:**
  - Behandelt die Eingabe als rohen Eingabestrom (einschließlich des Leerzeichens in der Menge der zu verwendenden Zeichen)
  - Verwendet den BPE- oder Unigramm-Algorithmus, um das entsprechende Vokabular zu konstruieren

- Normalisierer
  - Weist jedem Unicode-Zeichen semantisch äquivalente Zeichen aus verschiedenen Schriftsystemen zu
- Trainer
  - Spezialisiert darauf, ein Modell für Subwort-Tokenisierung zu erlernen, z.B. Unigramm
- Encoder
  - Nutzt die Normalisierungs- und Trainingseinheiten, um eingegebene Textfolgen in Subwörter zu segmentieren
- Decoder
  - Rekonstruiert die ursprünglichen Texte sprach- und schriftunabhängig

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Linguistisch motivierte Tokenisierungsansätze

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Linguistisch motivierte Tokenisierer Motivation[11],[12] und morphologische Analysatoren/Werkzeuge

- Bisherige Tokenisierer:
  - verlassen sich auf die statistischen Eigenschaften des Korpus
  - ignorieren das in der Sprache eingebettete linguistische Wissen

- Behauptung:
  Ein morphologisch informiertes Vokabular führt zu einer besseren Generalisierungsfähigkeit von Sprachmodellen.

- **Finite-State-Transducer (FST)[13]:** lexikalisch akzeptierte Formen werden mithilfe regulärer Ausdrücke in einen FST überführt.
  Ermöglicht die Speicherung einer Sammlung aller zulässigen Token, einschließlich Satzzeichen und spezieller Symbole, an festgelegten Positionen.

- **Morfessor 2.0[14]**: Familie von probabilistischen maschinellen Lernmethoden zur Ermittlung der morphologischen Segmentierung von Rohtextdaten.

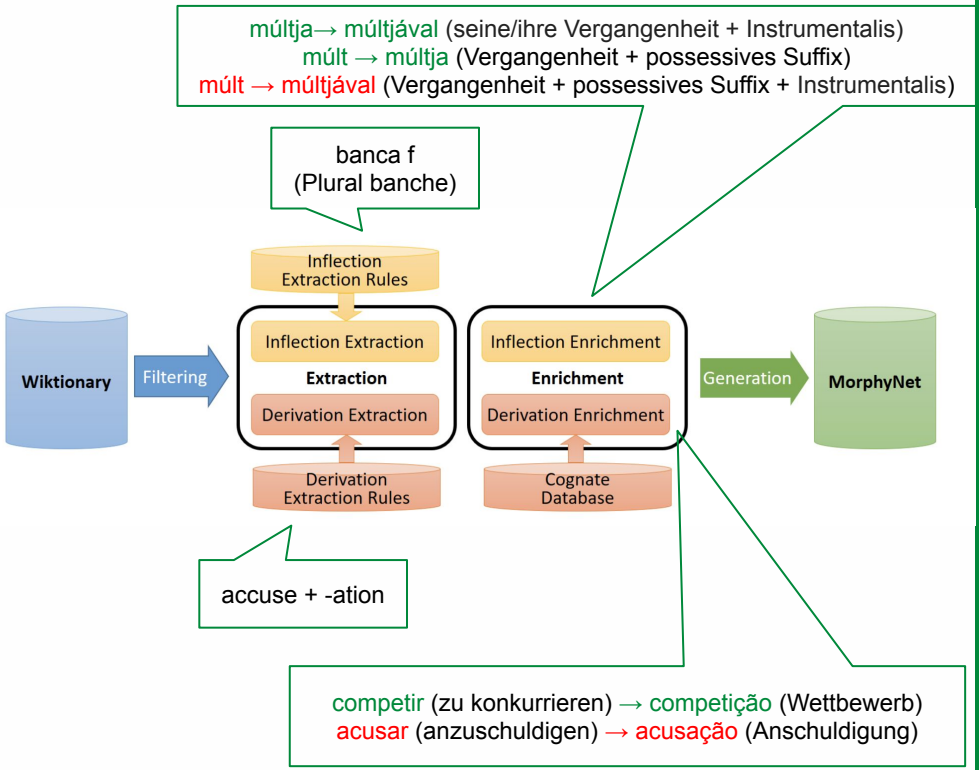
- **MorphemePiece[15]:** verwendet Nachschlagetabelle und WordPiece.

- **MorphPiece[16]:** verwendet Nachschlagetabelle und BPE.

- **Sprachspezifische Analysatoren:** z.B. für Türkisch → Zemberek[17]

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

MorphyNet[18]



MorphyNet: hochwertige morphologische Datenbank

- Flexionsdaten sind von besonderer Bedeutung für morphologisch reiche Sprachen
- Ableitungen nützlich, z. B. um die Semantik von Wörtern außerhalb des Vokabulars abzuleiten
- deckt 15 Sprachen ab
- 519k Derivations- und 10,1M Flexionseinträgen und eine Vielzahl von morphologischen Merkmalen

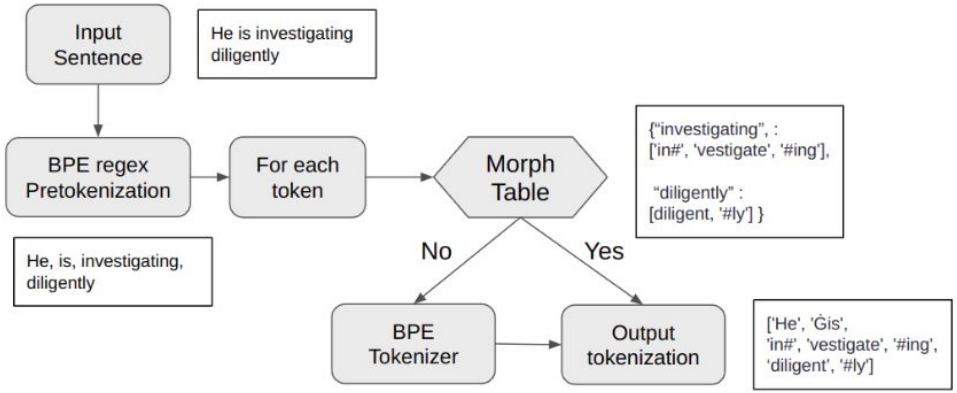
Prozess:

1. **Filterung:** liefert Inhalte aus bestimmten Abschnitten relevanter lexikalischer Einträge: Schlagwortzeilen, Etymologieabschnitte und Flexionstabellen werden für Substantive, Verben und Adjektive zurückgegeben.
2. **Extraktion:** erhält morphologische Rohdaten durch Parsing der obigen Abschnitte.
3. **Bereicherung:** erweitert algorithmisch die Abdeckung der Ableitungen und Flexionen von Wiktionary, durch völlig unterschiedliche Methoden
4. Die **Ressourcengenerierung** schließlich gibt MorphyNet-Daten aus.

Abbildung angelenht an Batsuren et al. 2021

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

MorphPiece[16]



- MorphPiece:
  - Ein Tokenisierungsschema für die englische Sprache
  - Kombiniert das BPE mit morphembasierter Segmentierung
  - Führt zu einem stärkeren linguistisch ausgerichteten Tokenisierungsmechanismus

- Prozess:
  - Text wird normalisiert und vortokenisiert gemäß dem BPE-Standard.
  - Vortoken werden durch eine Nachschlagetabelle von Wörtern (MorphTable genannt) geleitet
    - morphembasierte Segmentierung verfügbar? → Pretoken durch die entsprechenden Morpheme ersetzt
    - sonst nach BPE mit einem speziell trainierten Vokabular tokenisiert

Abbildung: Jabbar H. 2023

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

MorphPiece[16]

Word	BPE tokens	Wordpiece tokens	MorphPiece tokens
batting	'bat', 'ting'	batting	'bat', '#ing'
disengage	'dis', 'eng', 'age'	'di', '##sen', '##ga', '##ge'	'dis#', 'en#', 'gage'
archeologists	'ar', 'che', 'ologists'	'arch', '##eo', '##logists'	'archaeo#', '#logy', '#ist', '#s'
decompress	'dec', 'omp', 'ress'	'deco', '##mp', '##ress'	'de#', 'compress'
photographers	'phot', 'ographers'	photographers	'photo#', '#graph', '#er', '#s'

- MorphPiece
  - linguistisch ausgerichtete Affixe mit semantische Bedeutung
  - ändert die Schreibweise einiger Wörter, ohne die diese Segmentierung nicht möglich wäre (['bat','ing'] statt ['batt','ing'])
  - Negationspräfixe wie 'de', 'un' und 'dis' klar vom Stamm getrennt

- MorphGPT übertrifft in den meisten Fällen die GPT-Version sowohl bei LM- als auch bei NLU-Aufgaben

Sprachmodellierung Aufgaben

Dataset	Metric	GPT-2 50k	GPT-2 Base	Morph 50k	Morph 100k	Morph 150k	Morph 200k	GPT-2 Large
PennTreeBank	ppl	79.31	61.58	43.2	39.85	38.74	38.25	37.94
OpenAI-250K	ppl	30.0	25.58	18.74	17.89	17.47	17.26	16.74
Lambada	ppl	74.97	55.78	47.11	45.38	43.25	42.83	37.21
Lambada	acc	0.44	0.468	0.556	0.567	0.584	0.586	0.593

NLU Aufgaben

Task	GPT-2	MorphGPT	Difference
RTE	0.6318	**0.7004**	**10.86%**
SST	0.9163	**0.9209**	**0.50%**
QQP	**0.8981**	0.8913	-0.76%
WNLI	0.3662	**0.4648**	**26.93%**
MRPC	0.7402	**0.8015**	**8.28%**
COLA	0.2574	**0.4542**	**76.46%**
QNLI	**0.8772**	0.8766	-0.07%
MNLI	**0.8216**	0.8167	-0.60%
AVG	68.86	**74.08**	7.58

Abbildungen: Jabbar H. 2023

Morphologischer Tokenisierer für Türkisch[19]

Toplumsal barış sağlanır → Der soziale Frieden ist erreicht

- Toplum (Gesellschaft), -sal (Derivationsuffix: N auf Adj),
- Barış (Frieden),
- Sağla- (zu beschaffen), -n (Passivmarkierung), -ır (Tempusmarker: Präsens)

Method	Tokenized text
Character-level	"t", "o", "p", "l", "u", "m", "s", "a", "l", " ", "b", "a", "r", "ı", "ş", " ", "s", "a", "ğ", "l", "a", "n", "ı", "r"
BPE	"[CLS]", "toplumsal", "barış", "sağ", "##lanır", "[SEP]"
WordPiece	"[CLS]", "toplumsal", "barış", "sağlan", "##ır", "[SEP]"
Morphological-level	"[CLS]", "toplum", "##sal", "barış", "sağ", "##lanır", "[SEP]"
Word-level	"[CLS]", "[UNK]", "barış", "[UNK]", "[SEP]"

		News Classification			Hate Speech Detection			Sentiment Analysis			Named Entity Recognition			Semantic Text Similarity		Natural Language Inference		
		P	R	F1	P	R	F1	P	R	F1	P	R	F1	corr	p-value	P	R	F1
	BERT	**0.918**	**0.917**	**0.917**	**0.781**	**0.781**	**0.781**	**0.927**	**0.927**	**0.927**	**0.935**	**0.955**	**0.945**	**0.862**	**<1e-178**	**0.852**	**0.852**	**0.852**
R-TR-medium	Char	0.715	0.723	0.713	0.606	0.609	0.607	0.812	0.812	0.812	0.730	0.788	0.757	0.256	<1e-4	0.620	0.619	0.619
	BPE	**0.886**	**0.885**	**0.885** •	0.742	0.737	0.738	0.882	0.881	0.881 ◦	0.851	0.883	0.866 ◦	0.487	<2e-32	0.772	0.772	0.772
	WP	0.882	0.881	0.881 ◦	**0.745**	**0.745**	**0.745** •	**0.884**	**0.884**	**0.884** •	**0.858**	**0.893**	**0.875** •	**0.718**	<3e-92 •	**0.778**	**0.778**	**0.778** •
	Morph	0.869	0.868	0.867	0.726	0.727	0.726	0.824	0.823	0.823	0.839	0.872	0.855	0.655	<5e-63 ◦	0.768	0.768	0.768
	Word	0.857	0.857	0.856	0.647	0.649	0.648	0.805	0.805	0.805	0.791	0.740	0.764	0.492	<2e-16	0.603	0.598	0.595

- Morphologischer Tokenisierer:
  - Verwendung von morphologischem Analysetool Zemberek[17] für Türkisch
  - Vorteil: Semantik auf der Grundlage der Wortsuffixe zu lernen
  - Nachteil: Wortstämme können nicht weiter aufgespalten werden → Vokabulargröße erhöht
- Experimente:
  - RoBERTa-TR-medium, mit verschiedenen Tokenisierungsalgorithmen und unterschiedlichen Vokabulargrößen
  - Morphologischer Tokenisierer konkurrenzfähig mit BPE und WordPiece
  - Mit Vergrößerung des Vokabulars, die Leistung vom morphologischen Tokenisierer stärker verbessert als die von BPE und WordPiece

Abbildungen: Toraman et al. 2023

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Fazit - Kritik

Es gibt keine perfekte Methode für die Tokenisierung.

Gruppe	Effizienz im Vokabular	Umgang mit seltenen Wörtern	Linguistisches Feingefühl	Sprachunabhängigkeit	Besonders geeignet für	Anforderungen an die Trainingsdaten	Rechnerische Effizienz	Abwägung zwischen Genauigkeit und Effizienz	Zukünftige Richtungen
Statistisch motiviert	Effizient beim Aufbau eines überschaubaren Vokabulars	Verringert bzw. schließt OOV Wörter aus	Möglicherweise werden sprachliche Strukturen wie die Morphologie nicht beachtet	Sprach- und schriftunabhängig	Keine besondere Eignung für Sprachen in Bezug auf die morphologische Typologie (gleich gut)	Sind trainiert mit Rohtextdaten, wodurch sie besser skalierbar und zugänglich sind	Im Allgemeinen rechnerisch effizienter	Rechnerische Effizienz	Konvergenz von statistischen und linguistischen Ansätzen, wobei die Stärken beider Ansätze genutzt werden
Linguistisch motiviert	Erhöht Vokabulargröße	Haben Schwierigkeiten mit Wörtern, die nicht in ihren Sprachregeln enthalten sind	Sind darauf ausgerichtet, sprachliche Strukturen wie die Morphologie zu verstehen und zu erhalten.	Muss für jede Sprache neu definiert werden	Morphologisch reiche Sprachen (z.B. agglutinierende: Türkisch)	Häufig erfordern annotierte Daten und linguistisches Fachwissen für das Training	Häufig erfordern eine komplexe linguistische Analyse	Sprachliche Genauigkeit	

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Literatur

1. Mielke, Sabrina J., et al. "Between words and characters: a brief history of open-vocabulary modeling and tokenization in nlp." arXiv preprint arXiv:2112.10508 (2021).
2. Hugging Face: Tokenization Overview and Summary. Website: https://huggingface.co/docs/transformers/tokenizer_summary
3. Hugging Face: Normalization and Pre-tokenization. Website: https://huggingface.co/learn/nlp-course/chapter6/4
4. Schuster, Mike, and Kaisuke Nakajima. "Japanese and korean voice search." 2012 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, 2012.
5. Song, Xinying, et al. "Fast wordpiece tokenization." arXiv preprint arXiv:2012.15524 (2020).
6. Sennrich, Rico, Barry Haddow, and Alexandra Birch. "Neural machine translation of rare words with subword units." arXiv preprint arXiv:1508.07909 (2015).
7. Gage, Philip. "A new algorithm for data compression." C Users Journal 12.2 (1994): 23-38.
8. Wang, Changhan, Kyunghyun Cho, and Jiatao Gu. "Neural machine translation with byte-level subwords." Proceedings of the AAAI conference on artificial intelligence. Vol. 34. No. 05. 2020.
9. Kudo, Taku. "Subword regularization: Improving neural network translation models with multiple subword candidates." arXiv preprint arXiv:1804.10959 (2018).
10. Kudo, Taku, and John Richardson. "Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing." arXiv preprint arXiv:1808.06226 (2018).

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Literatur

11. Hofmann, Valentin, Janet B. Pierrehumbert, and Hinrich Schütze. "DagoBERT: Generating derivational morphology with a pretrained language model." arXiv preprint arXiv:2005.00672 (2020).

12. Hofmann, Valentin, Janet B. Pierrehumbert, and Hinrich Schütze. "Superbizarre Is Not Superb: Derivational Morphology Improves BERT's Interpretation of Complex Words." arXiv preprint arXiv:2101.00403 (2021).

13. Karttunen, Lauri, et al. "Regular expressions for language engineering." Natural Language Engineering 2.4 (1996): 305-328.

14. Smit, Peter, et al. "Morfessor 2.0: Toolkit for statistical morphological segmentation." The 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL), Gothenburg, Sweden, April 26-30, 2014. Aalto University, 2014.

15. Bratt, Jonathan. “MorphemePiece”. GitHub website: https://github.com/macmillancontentscience/morphemepiece

16. Jabbar, Haris. "MorphPiece: Moving away from Statistical Language Representation." arXiv preprint arXiv:2307.07262 (2023).

17. Akın, Ahmet Afsin, and Mehmet Dündar Akın. "Zemberek, an open source NLP framework for Turkic languages." Structure 10.2007 (2007): 1-5.

18. Batsuren, Khuyagbaatar, Gábor Bella, and Fausto Giunchiglia. "Morphynet: a large multilingual database of derivational and inflectional morphology." Proceedings of the 18th sigmorphon workshop on computational research in phonetics, phonology, and morphology. 2021.

19. Toraman, Cagri, et al. "Impact of tokenization on language models: An analysis for turkish." ACM Transactions on Asian and Low-Resource Language Information Processing 22.4 (2023): 1-21.

20. Alyafeai, Zaid, et al. "Evaluating various tokenizers for Arabic text classification." Neural Processing Letters 55.3 (2023): 2911-2933.

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Literatur

21. Schwartz, Lane, et al. "Neural polysynthetic language modelling." arXiv preprint arXiv:2005.05477 (2020).

22. Pan, Yirong, et al. "Morphological word segmentation on agglutinative languages for neural machine translation." arXiv preprint arXiv:2001.01589 (2020).

23. Matthews, Austin, Graham Neubig, and Chris Dyer. "Using morphological knowledge in open-vocabulary neural language models." Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). 2018.

24. Ataman, Duygu, and Marcello Federico. "An evaluation of two vocabulary reduction methods for neural machine translation." Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track). 2018.

25. Zhou, Giulio. Morphological zero-shot neural machine translation. Diss. Master's thesis, University of Edinburgh, 2018.

26. Domingo, Miguel, et al. "How much does tokenization affect neural machine translation?." International Conference on Computational Linguistics and Intelligent Text Processing. Cham: Springer Nature Switzerland, 2019.

27. Macháček, Dominik, Jonáš Vidra, and Ondřej Bojar. "Morphological and language-agnostic word segmentation for nmt." International Conference on Text, Speech, and Dialogue. Cham: Springer International Publishing, 2018.

28. Sälevä, Jonne, and Constantine Lignos. "The effectiveness of morphology-aware segmentation in low-resource neural machine translation." arXiv preprint arXiv:2103.11189 (2021).

29. Banerjee, Tamali, and Pushpak Bhattacharyya. "Meaningless yet meaningful: Morphology grounded subword-level NMT." Proceedings of the second workshop on subword/character level models. 2018.

30. Huck, Matthias, Simon Riess, and Alexander Fraser. "Target-side word segmentation strategies for neural machine translation." Proceedings of the Second Conference on Machine Translation. 2017.

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Appendix

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Verwandte Arbeiten[1]

- Alyafeai et. al. (2023)[20] stellen in drei Klassifizierungsaufgaben auf Arabisch heraus, dass Datensätze mit sehr wenig Ressourcen von Informationen über die Morphologie und der großen Tokengröße profitieren.
- Die Low-Resource-Studie von Schwartz et al. (2020)[21] zeigt, dass Morfessor-basierte Sprachmodelle die BPE-basierten übertreffen.
- Pan et al. (2020)[22] verbessern ebenfalls die NMT für Türkisch und Uigurisch, indem sie morphologische Analysatoren vor der Anwendung von BPE einsetzen.
- Matthews et al. (2018)[23] zeigen, dass die Sprachmodellierung für agglutinierende Sprachen verbessert werden kann, wenn (manuell analysierte) morphologische Analysen verwendet werden.
- Ataman und Federico (2018b)[24] finden dass unter Verwendung von unüberwacht gewonnenen "morphologischen" Teilwörtern, ein auf Morfessor FlatCat basierendes Modell BPE übertreffen kann.
- Zhou (2018)[25], Domingo et al. (2018)[26], Macháček et al. (2018)[27] und Saleva und Lignos (2021)[28] finden keine zuverlässige Verbesserung gegenüber BPE für die Übersetzung.
- Banerjee und Bhattacharyya (2018)[29] analysieren Übersetzungen, die mit Morfessor und BPE segmentiert wurden, und kommen zu dem Schluss, dass eine mögliche Verbesserung von der Ähnlichkeit der Sprachen abhängt.
- Huck et al. (2017)[30] schlagen daher vor, beide Ansätze zu kombinieren.

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Statistiken zum MorphyNet-Datensatz[18]

		Inflectional morphology			Derivational morphology			
#	Languages	words	entries	morphemes	words	entries	morphemes	Total
1	Finnish	65,402	1,617,751	1,139	18,142	37,199	446	1,654,950
2	Serbo-Croatian	68,757	1,760,095	263	8,553	20,008	429	1,780,103
3	Italian	75,089	748,321	104	22,650	42,149	749	790,470
4	Hungarian	38,067	1,034,317	428	14,566	37,940	832	1,072,257
5	Russian	67,695	1,343,760	252	21,922	36,922	575	1,380,682
6	Spanish	67,796	677,423	145	16,268	27,633	490	705,056
7	French	44,729	453,229	98	15,473	37,203	636	490,432
8	Portuguese	30,969	329,861	161	10,504	15,974	387	345,835
9	Polish	36,940	663,545	251	9,518	18,404	405	681,949
10	German	35,086	214,401	243	13,070	23,867	465	238,268
11	Czech	9,781	298,888	112	4,875	9,660	318	307,935
12	English	149,265	652,487	8	67,412	200,365	2,445	852,852
13	Catalan	16,404	168,462	91	3,244	4,083	220	172,545
14	Swedish	14,485	131,693	32	3,190	5,810	217	137,503
15	Mongolian	2,085	14,592	35	1,410	1,940	229	16,532
	Total	722,550	10,108,825	3,362	230,797	519,157	8,843	10,627,369

 Abbildung: Batsuren et al. 2021

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Vielen Dank für Ihre Aufmerksamkeit!

Abbildung: unsplash.com