Entwicklung eines lexikalischen Verfahrens auf der Basis endlicher Automaten an der ZBW

Moritz Fürneisen, Christopher Bartz, Anna Kasprzik
ZBW – Leibniz-Informationszentrum Wirtschaft
DNB Annif Workshop, 3./4. Dezember 2020

ZBW Leibniz-Informationszentrum Wirtschaft
Leibniz Information Centre for Economics

Die ZBW ist Mitglied der Leibniz-Gemeinschaft.

Überblick

- assoziative und lexikalische Verfahren
- Maui und stwfsa
- Endliche Automaten
- Vorgehen stwfsa
- Metrikwerte STWFSA
- Herausforderungen bei Implementierung
- Zukunftspläne STWFSA

Vorgehen Lexikalische Verfahren

1. Konzepte anhand ihrer Labels im Text finden
2. gefundene Konzepte bewerten

Mögliche Bewertungskriterien:

- Position der Fundstelle
- Häufigkeit des Konzeptes im Gesamttext
- Verortung des Konzepts im Vokabular

Vorgehen Assoziative Verfahren

1. Gesamten Eingabe in Featurevektor transformieren
2. Training mit Paaren von Featurevektoren und Menge vergebener Konzepte
3. Unbekannte Eingabe wird in Featurevektor transformiert und anhand dieser verschlagwortet

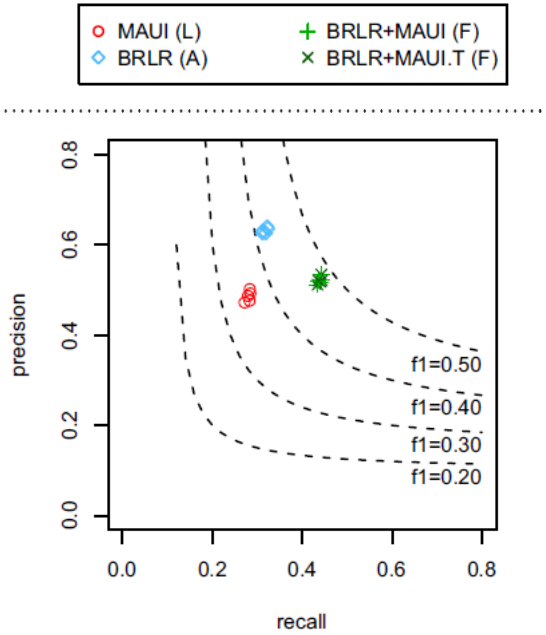
Lexikalisch vs. Assoziativ

	Aspect	L	A
A1	Amount of required training data	++	−
A2	Prediction of unseen concepts	++	−−
A3	Prediction of synonyms	−−	++
A4	Ambiguity	o	+
A5	Exploitation of thesaurus relations	+	o

Toepfer, M., & Seifert, C. (2018). Fusion architectures for automatic subject indexing under concept drift. *International Journal on Digital Libraries*, *21*(2), 169-189.

Direkter Vergleich

- lexikalische Verfahren nach Metrikwerten meist schlechter
- liefert häufig andere Ergebnisse
- In Kombination bessere Ergebnisse als alleine



Toepfer, M., & Seifert, C. (2018). Fusion architectures for automatic subject indexing under concept drift. *International Journal on Digital Libraries*, *21*(2), 169-189.

Maui: Das lexikalische Backend in Annif

- In Java programmiert
- Wird in Annif per REST API eingebunden
  - Dadurch im Vergleich zu anderen Verfahren erschwerte Konfiguration
  - Geschwindigkeitsverlust durch Serialisierung und Kommunikation
- Letztes Update: Oktober 2015

STWFSA: Lexikalisches Verfahren der ZBW

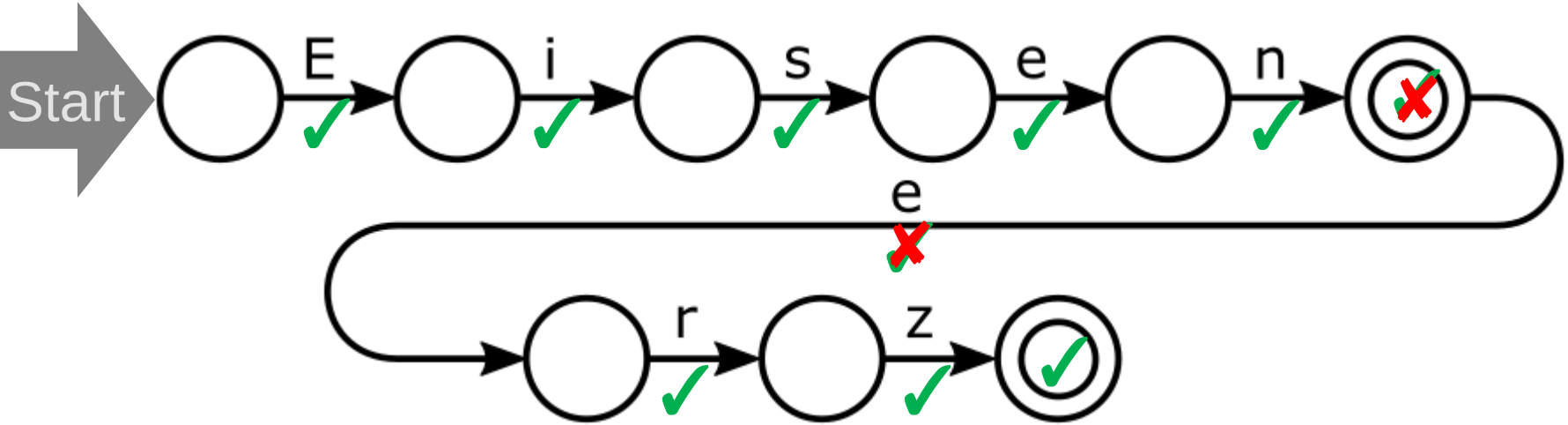
- Angepasst an den STW und die Metadaten der ZBW
- In Java und Python programmiert
  - Java: Konzepte finden
  - Python: gefundene Konzepte bewerten
  - REST Schnittstelle zwischen beiden Komponenten

Auffinden von Konzepten

- Abgleich mit allen Bezeichnungen zeitaufwändig
- STWFSA nutzt gemeinsame Präfixe der Wörter

Endlicher Automat Eisen(erz)?

Eingabe: Eisenstange



STWFSA: Vorgehen

1. Extrahiere Konzept-Bezeichnerpaare aus Thesaurus
2. Extrahiere relevante Teile aus Bezeichnung

   BIP (Bruttoinlandsprodukt) -> BIP -> B.?I.?.P.?
3. Erstelle endlichen Automaten

   Endzustände entsprechen Konzepten

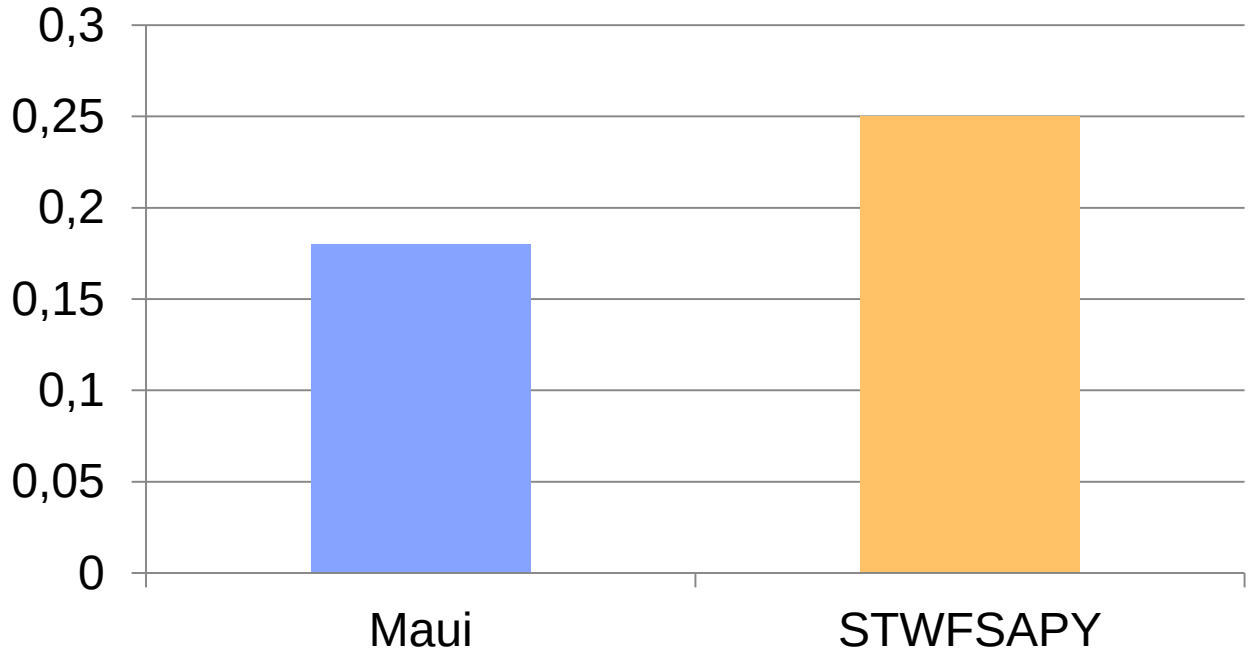
Auffinden von Konzepten in Text

- Überprüfe für jeden Suffix ob der Automat in einen Endzustand gelangt

Experimente

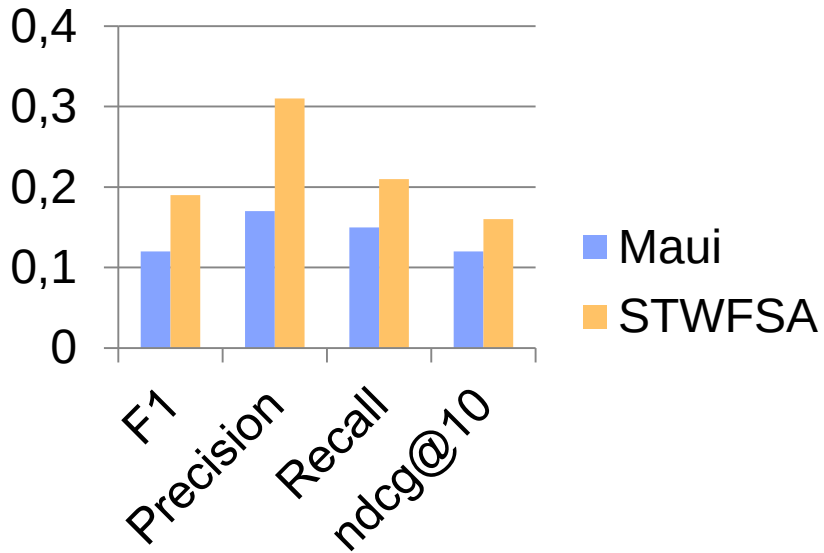
- Vergleich mit Maui
- Auf Titeln und Autorenkeywords
- Datensätze
  - Econis
  - Annif tutorial
- Sprache Englisch

Econis: F1 Werte über Dokumente gemittelt

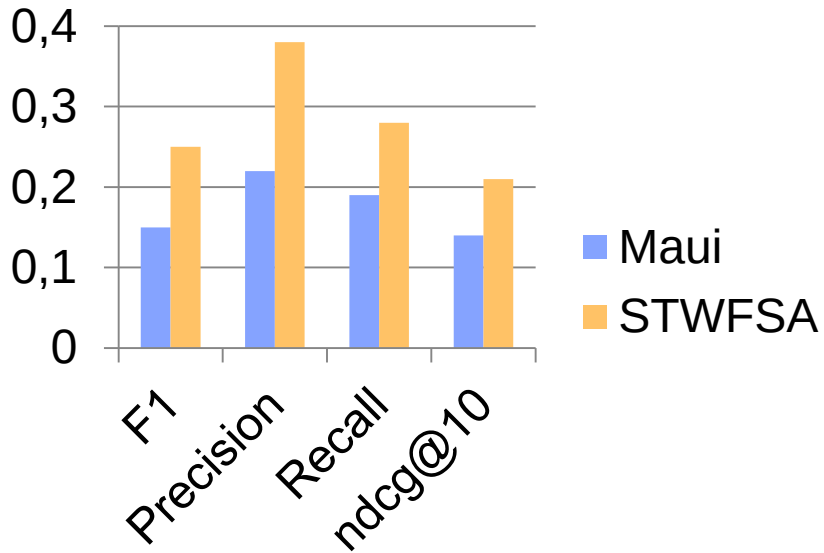


Metriken für Annif Tutorial Datensätze

YSO-Finna



STW-Econbiz



Herausforderungen bei Implementierung

- kein passender Ersatz für Java-Library zur Erstellung des Automaten, der die regulären Ausdrücke findet
  - existierende Automatenlibraries können nur sagen, dass ein Konzept gefunden wird, aber nicht, welches

- daher: Implementierung von Automatenerstellung und Suchalgorithmus durch AutoSE@ZBW
  - Weitaus höherer Aufwand als geplant

Ergebnisse

- Python-Library zur Suche und Bewertung von Thesauruskonzepten
  https://github.com/zbw/stwfsapy
- Einbindung in Annif geplant
  - Bis dahin https://github.com/mo-fu/annif
- bessere Resultate als *Maui in der Standardkonfiguration in Annif*

Zukünftige Entwicklungen

- Unterstützung weiterer Sprachen
- Verbesserung der Ergebnisse auf längeren Texten
- Betrachtung von Nachbarn der gefundenen Konzepte
- Enfernen von Homonymzusätzen: z.B.: Lead (Metal) -> Lead

Backup: Hash Funktionen

- große Menge von möglichen Eingabewerten auf kleine Menge von Zielwerten (0,1, …, n) abbilden
- Alle Eingaben mit gleichem Zielwert werden zusammengefasst

Überprüfung von Existenz eines Elements e in Menge A

1. Für alle Elemente a in A speichere a an Listenposition hash(a)
2. Für Element e überprüfe ob zwischen e und einem der Elemente an Listenposition hash(e) Gleichheit besteht

- Sinnvoll, wenn die Existenz in A häufig überprüft werden soll

Backup: Maui

Indexierung

1. Normalisiere alle Konzeptbezeichnungen (Endungen entfernen)
2. Berechne Hashwerte für Bezeichnungen

Konzepterkennung

1. Normalisiere Eingabetext auf gleiche Weise wie bei Indexerstellung
2. Für alle Wortketten *wk* der Länge *1 bis l:*

    Nutze hash(wk) um Existenz von Konzepten zu überprüfen