

# EINE EINFÜHRUNG IN DAS LANGUAGE TOOL

## Einleitung

Dieses Dokument soll eine Einführung in das LanguageTool geben. Dabei soll vor allem auf die Hintergründe für die Entwicklung eingegangen werden, sowie die generelle Funktionsweise. Relevante Informationen zur Syntax, eine detaillierte Beschreibung der vorhandenen Regeln und Testmodule für die Anwendung sind nicht in dieser Einführung dokumentiert, können aber im Arbeitsbereich LanguageTool auf der Kursseite Texttechnologie abgerufen werden.

## Einführung in das LanguageTool

Das LanguageTool ist eine kostenlose, Open-Source Applikation zur Überprüfung der Rechtschreibung, Grammatik und des Schreibstils von Texten. Dabei werden einzelne Wörter, Wortteile und ganze Sätze auf Übereinstimmung mit verschiedenen Regeln im Hintergrund des Systems verglichen. Diese Regeln beschreiben typische Fehler in verschiedenen Kategorien. Sobald ein Fehler erkannt wird, wird die entsprechende Textpassage markiert. Je nach Anwendung kann dann eine Fehlermeldung angezeigt werden, die zusätzliche Informationen zu dem Fehler ausgibt, siehe Abbildung 1.

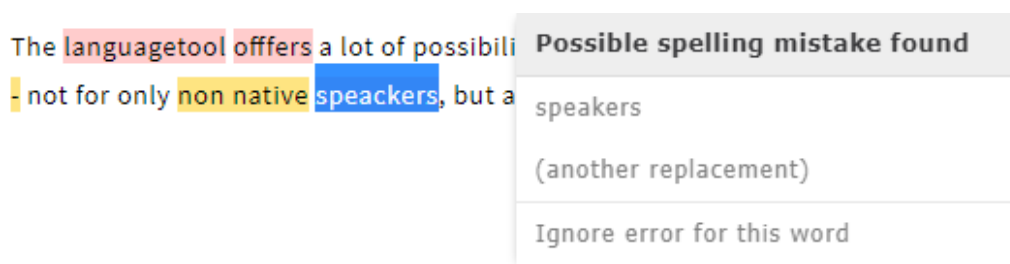


Abbildung 1 Ausgabe eines Fehlers im LanguageTool<sup>1</sup>

So kann der Nutzer nicht nur den Fehler erkennen, sondern kann auch eine Begründung für die Fehlerausgabe abrufen und, falls verfügbar, kann der Fehler auch direkt verbessert werden. Das Tool ist daher für verschiedene Gruppen anwendbar. Es hilft sowohl Muttersprachlern als auch Nicht-Muttersprachlern beim Erkennen von Rechtschreib- und Grammatikfehlern und unterstützt einen ansprechenden Schreibstil. Das LanguageTool unterstützt derzeit 31 Sprachen. Darunter Deutsch, Englisch und Spanisch, sowie verschiedene Dialekte. So kann beispielsweise auch Rechtschreibung, Grammatik und Stil für Schweizer Deutsch abgefragt werden.

## Hintergründe zur Entwicklung des LanguageTools

Das LanguageTool wurde von Daniel Naber und Marcin Milowski im August 2005 veröffentlicht. Seither wird das Tool stets weiterentwickelt. Die Idee für das LanguageTool entstand durch die Diplomarbeit von Daniel Naber: „A Rule-Based Style and Grammar Checker“ an der Universität Bielefeld im Jahr 2003. Das Ziel dieser Arbeit war die Entwicklung einer Open-Source Applikation, die nicht nur eine Rechtschreibprüfung durchführt, sondern auch eine Grammatik und Stilprüfung. Darin beschreibt er die Entwicklung einer Applikation zur regelbasierten Stil- und Grammatikprüfung für die englische Sprache. Die Applikation sollte kostenlos sein und sowohl mit einem Text-Editor, als auch für sich

---

<sup>1</sup> LanguageTool GmbH

alleine angewendet werden können. Die Applikation sollte vor allem zwei Arten von Fehlern erkennen. Zum einen ungrammatische Konstruktionen, die dadurch entstehen, dass der Textproduzent eine Grammatikregel nicht kennt oder falsch benutzt und zum anderen Tippfehler und Verwechslungen von Wörtern, die zu einer falschen Grammatik führen, aber von der Rechtschreibkontrolle nicht erkannt werden. Dazu gehören beispielsweise no / now, their / there und is / it. Tippfehler, die schon von einer Rechtschreibkontrolle als Fehler erkannt werden, spielen für die Grammatikprüfung keine Rolle, da sie von einem Text ausgeht, der bereits mit der Rechtschreibkontrolle geprüft wurde.<sup>2</sup>

## Anwendung des LanguageTools

Das LanguageTool kann sowohl als Browser Extension als auch in einem Text-Editor angewendet werden. Für die Textprüfung im Browser muss die entsprechende Browser-Extension installiert und aktiviert werden. Die Anwendung des LanguageTools im Browser kann beispielsweise für das Überprüfen von E-Mail-Nachrichten oder Forum-Beiträgen genutzt werden. Dabei muss der entsprechende Text markiert und danach das Icon im Browserfenster ausgewählt werden. Das Tool vergleicht dann den Text mit den vordefinierten Regeln und markiert die Stellen, die einen Fehler aufzeigen. Außerdem wird eine Liste der Fehler angezeigt. Die Fehlermeldungen können dann ausgewählt werden, um mehr Informationen über den Fehler zu erhalten. Falls vorhanden können direkt Verbesserungen ausgewählt und eingesetzt werden.

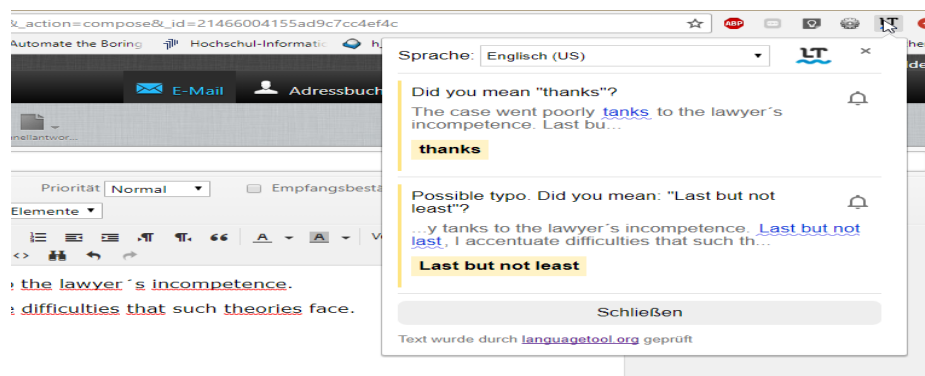


Abbildung 2 Fehlererkennung im Browser

Das LanguageTool kann auf eine ähnliche Art und Weise im Text-Editor verwendet werden. Dabei muss das LanguageTool heruntergeladen werden und in Microsoft Office oder Latex integriert werden. Im Editor ist anschließend ein neues Symbol in der Home Toolbar verfügbar, um den Text zu überprüfen. Dieses Symbol kann betätigt werden, wann immer der Nutzer seinen Text überprüfen möchte. Wie im Browser kann auch hier eine Liste mit den Fehlern angezeigt werden und Verbesserungen direkt ausgewählt werden, damit sich das Wort im Text automatisch an die Verbesserung anpasst und die Fehlermeldung ausgeblendet wird.

---

<sup>2</sup> Vgl. Daniel Naber 2003 S. 3

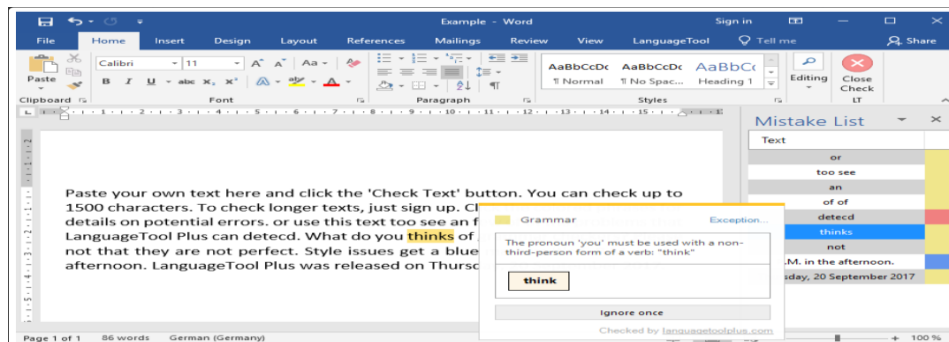


Abbildung 3 Fehlererkennung im Texteditor

## Funktionsweise des LanguageTools

Der folgende Abschnitt beschreibt die Funktionsweise des LanguageTools. Dabei werden zuerst die generellen Vorgaben an das LanguageTool definiert. Daraufhin wird der Prozess erläutert, um die Fehler zu erkennen sowie die notwendigen Tools.

### 1. Vorgaben an das LanguageTool

Das LanguageTool wurde entwickelt, um Menschen die Möglichkeit zu geben Dokumente mit weniger Fehlern zu schreiben. Ziel ist es, dass das Tool möglichst alle Fehler in Texten findet und ausgibt. Damit der Nutzer interaktiv mit dem Tool arbeiten kann, ist es wichtig, dass Fehler direkt erkannt und ausgegeben werden, damit der Nutzer diese direkt verbessern kann. Zusätzlich sollte eine falsche Fehlerausgabe ausgeschlossen werden, um die Glaubwürdigkeit an das Tool nicht einzuschränken und keine Fehler auszugeben, die keine Fehler sind. Außerdem soll das Tool für die verschiedenen Zwecke und Zielgruppen angepasst werden können.<sup>3</sup>

### 2. Notwendige Tools

Die Überprüfung nach Fehlern erfolgt aufgrund verschiedener vordefinierter Regeln im Hintergrund des Tools. Um Fehler zu erkennen, muss jedem Wort in der entsprechenden Textpassage das entsprechende Part-of-Speech Tag hinzugefügt werden und jeder Satz muss entsprechend in einzelne Chunks aufgesplittet werden. Somit können Sätze, Wortkombinationen und einzelne Wörter mit den Regeln verglichen werden und sobald eine Übereinstimmung besteht; als Fehler angezeigt werden. Zur Überprüfung des Texts werden verschiedene Hilfsmittel benötigt. Diese Hilfsmittel sind ein sogenannter Part-of-Speech Tagger und ein Chunker. „Unter "Tagging" versteht man in der Computerlinguistik allgemein die Annotation von Korpora mit linguistischen Informationen.“<sup>4</sup> Die bekanntesten Part-of-Speech Tags sind Nomen, Verb, Adjektiv oder Adverb. Zusätzlich können Nomen beispielsweise als Singular oder Plural definiert werden. Da ein Wort oft verschiedene POS-Tags haben kann, ist es oft schwierig für den Algorithmus den passenden Tag zu erkennen. Daher gibt es einen Disambiguator, der versucht durch verschiedene vordefinierte Regeln auf Basis der Semantik, der Wortstellung oder aufgrund von Häufigkeiten das entsprechende POS-Tag auszuwählen. Beim Chunking wird im Gegensatz zum POS-Tagging einer Sequenz von Wörtern ein Tag zugeordnet. Typische Tags können beispielsweise eine Nominalphrase oder eine Verbphrase sein. Eine

<sup>3</sup> Vgl. Daniel Naber 2003, S. 4

<sup>4</sup> Helmut Schmid

Nominalphrase Phrase besteht beispielsweise aus einem Nomen ausschließlich oder zusätzlich des dazugehörigen Adjektiv und Artikel.<sup>5</sup>

```
Disambiguator log:
JJ_NN_JJ[1]: bad[bad/JJ,bad/NN:U,I-NP-singular] -> bad[bad/JJ,I-NP-singular]
DT_JJ_NN[1]: English[English/JJ,English/NNP,E-NP-singular] -> English[English/NNP,E-NP-singular]
DT_JJ_NN[1]: .[./.*,./SENT_END*,0] -> .[./SENT_END*,0]
```

Token	Lemma	Part-of-speech	Chunk
	-	SENT_START	
He	he	PRP	B-NP-singular E-NP-singular
complains	complain	VBZ	B-VP
about	about	RB IN RP	B-PP
my	my	PRP\$	B-NP-singular
bad	bad	JJ	I-NP-singular
English	English	NNP	E-NP-singular
.	.	SENT_END PARA_END	O

Abbildung 4 Beispiel für die Untersuchung eines Textes im LanguageTool mit Lemmatizer, POS-Tagger, Chunker und Disambiguator

Der Ansatz der regelbasierten Überprüfung unterstützt dabei, vordefinierte Fehler zu erkennen. Dazu müssen die Regeln vorher im System eingetragen werden. Ein großer Nachteil dabei ist, dass Fehler, die nicht definiert wurden, nicht erkannt werden. Bei einer großen Häufigkeit an möglichen Fehlern ist es sehr schwierig ein komplettes Regelset zu definieren. Jedoch kann das Regelset Schritt für Schritt aufgebaut werden, und aufgrund einer relativ verständlichen Syntax von den Nutzern erweitert werden. Außerdem können Fehlermeldungen integriert werden, die dem Nutzer die Möglichkeit geben, dessen Fehler besser zu verstehen und optional zu verbessern.

```
<rule id="BED_ENGLISH" name="Possible typo 'bed/bat(bad) English/...'">
  <pattern>
    <marker>
      <token regexp="yes">bed|bat</token>
    </marker>
    <token regexp="yes">English|attitude</token>
  </pattern>
  <message>Did you mean <suggestion>bad</suggestion>?</message>
  <url>http://some-server.org/the-bed-bad-error</url>
  <example type="correct">Sorry for my <marker>bad</marker> English.</example>
  <example correction="bad" type="incorrect">Sorry for my <marker>bed</marker> English.</example>
</rule>
```

Abbildung 5 Beispiel der XML Syntax für eine Regel für die Erkennung von bed/bad

### 3. Hilfreiche Funktionen des LanguageTools

Auf der Seite des LanguageTools wurde eine Seite für die Community eingerichtet. Diese Seite bietet eine Auflistung der Regeln des LanguageTools und hilft den Nutzer Texte zu analysieren und Regeln zu erstellen. In der Liste der Regeln sind alle enthaltenen Regel der integrierten Sprachen aufgelistet. Diese Regeln sind nach Kategorie sortiert und es kann der Sourcecode oder die XML Syntax angezeigt werden.

Liste der Regeln: <https://community.languagetool.org/rule/list?lang=en>.

Zusätzlich kann ein beliebiger Text durch die interne LanguageTool-Textanalyse überprüft werden. Wie bereits in Abbildung 4 gezeigt, werden hier Angaben zu der Lemma Version der Wörter gesetzt

<sup>5</sup> Vgl. Daniel Naber 2003, S. 6

und der entsprechende Part-Of-Speech Tag und Chunk angezeigt.

Textanalyse: <https://community.languagetool.org/analysis/index?lang=en>.

Ein weiterer Aspekt des LanguageTools ist, dass Regeln mit einer einfachen XML Syntax geschrieben werden können. So kann der Nutzer selber Regeln schreiben und diese implementieren oder auch an die Entwickler des Tools weitergeben. Mit Hilfe eines Regel-Editors können die Muster zur Fehlererkennung erstellt werden. Hierbei können über 3 verschiedene Modi genutzt werden. Es kann zum einen die Regel schrittweise erstellt werden oder eine bereits gegebene Syntax eingefügt werden, die entweder durch verschiedene Stufen weiterbearbeitet werden kann, oder im Experten Modus direkt gegen Beispiele in Wikipedia und Tatoeba gegengeprüft wird. Mehr Informationen zur Regelerstellung können im Kursbereich Texttechnologie im Reiter LanguageTool gefunden werden.

Regel-Editor: <https://community.languagetool.org/ruleEditor2/index?lang=de>.

Des Weiteren wird ein Wiki angeboten, in dem alle notwendigen Informationen für Entwickler vorhanden sind, die das LanguageTool weiterentwickeln möchten. Hier können alle möglichen Informationen zu der Entwicklung von Regeln abgerufen werden. Außerdem bietet es verschiedene Hilfestellung zu der Einbindung des LanguageTools und ein Forum, in dem Fragen beantwortet werden können.

Wiki: <http://wiki.languagetool.org/>

### Das LanguageTool für die Leichte Sprache

Das Prinzip des LanguageTool ist, dass in einer Sprache verschiedene Regeln angewendet werden, die die Wörter, Wortformen und Zusammensetzungen betrachtet. Daher können schwierige Wörter, Wortkombinationen, Zeitformen, Satzlängen und vieles mehr, einfach erkannt werden, sobald sie in den Regeln definiert wurden. Im Tool wurden bereits Regeln für Leichte Sprache erstellt und implementiert. In diesen Regeln wurden beispielsweise Warnungen implementiert, die den Nutzer bei der Verwendung von Passiv, Konjunktiv, zu langen Sätzen oder Metaphern warnt. Diese Regeln bieten einen guten und grundlegenden Ansatz zur Überprüfung von Texten aufgrund Leichter Sprache. Da diese Regeln allerdings nicht das komplette Spektrum an Regeln zur Leichten Sprache abdecken müssen weitere Regeln definiert und hinzugefügt werden.

Dies ist kein Problem, da das Open-Source Tool durch die verwendete XML Syntax einen einfachen Weg bietet, eigene Regeln zu erstellen. Durch die Community wird das Tool regelmäßig erweitert und kann somit auch mit Regeln zur Leichten Sprache erweitert werden. Verschiedenen Tutorials auf der Website des LanguageTools<sup>6</sup> geben einen Einblick in die Erstellung der Regeln für die verschiedenen Sprachen. Diese können je nach Belieben angepasst werden und durch Vertreter des Tools implementiert werden oder für eigene Zwecke in einer lokalen Version eingearbeitet und verwendet werden. Des Weiteren bietet das LanguageTool auf seiner Website eine Plattform, um Regeln zu erstellen.

---

<sup>6</sup> <https://community.languagetool.org/ruleEditor2/>

## Literaturverzeichnis

Daniel Naber (2003): A Rule-Based Style and Grammar Checker. Technische Fakultät, Universität Bielefeld. Bielefeld. Online verfügbar unter [http://www.danielnaber.de/language-tool/download/style\\_and\\_grammar\\_checker.pdf](http://www.danielnaber.de/language-tool/download/style_and_grammar_checker.pdf).

Helmut Schmid: Einführung. Hg. v. Sektion Computerlinguistik der Deutschen Gesellschaft für Sprachwissenschaft (DGfS). Online verfügbar unter <http://www.computerlinguistik.org/portal/portal.html?s=Tagging>, zuletzt geprüft am 29.07.2018.

LanguageTool GmbH (Hg.): LanguageTool Startseite. Online verfügbar unter <http://wiki.languagetool.org/>, zuletzt geprüft am 19.07.2018.

LanguageTool GmbH (B) (Hg.): LanguageTool Style and Grammar Checker Wiki. Online verfügbar unter <http://wiki.languagetool.org/>, zuletzt geprüft am 19.07.2018.