Using Web N-Grams to Help Second-Language Speakers

Martin Potthast     Martin Trenkmann     Benno Stein
Bauhaus-Universität Weimar
www.webis.de
Introduction
Introduction

Writing in a foreign language is difficult.

Problems include

- Spelling
- Grammar
- Translation
- Word Choice
- Writing Style

Tools include

- Spell checkers.
- Grammar checkers.
- Dictionaries, (machine translation).
- Thesauri.
- Style checkers.

Anything missing?
Introduction

What about text commonness?
Introduction

What about text commonness?

Correctness vs. Commonness

We present NETSPEAK, a tool

- to assist with word choice, and
- to check phrase commonness.

NETSPEAK implements wildcard queries on top of a Web n-gram index.
## NETSPEAK: The Writing Assistant

### Frequency and Phrases

<table>
<thead>
<tr>
<th>Frequency</th>
<th>Phrase</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>19,103</td>
<td>looks fine to me</td>
<td><img src="http://www.netspeak.cc" alt="Icon" /></td>
</tr>
<tr>
<td>810</td>
<td>looks fine for me</td>
<td><img src="http://www.netspeak.cc" alt="Icon" /></td>
</tr>
<tr>
<td>353</td>
<td>looks fine with me</td>
<td><img src="http://www.netspeak.cc" alt="Icon" /></td>
</tr>
<tr>
<td>107</td>
<td>looks fine by me</td>
<td><img src="http://www.netspeak.cc" alt="Icon" /></td>
</tr>
<tr>
<td>20,373</td>
<td>100.0 %</td>
<td>0.186 seconds</td>
</tr>
</tbody>
</table>

©2010 webis.de | Home | Learn more | Plugin | API | Terms | About | Permalink

http://www.netspeak.cc

6 Potthast at WEBNGRAM at SIGIR’10
Wildcard N-Gram Retrieval
Wildcard N-Gram Retrieval

Given a set of $n$-grams, $n \leq 5$, and their frequencies.

A query $q$ defines a pattern as a sequence of $n$-grams and wildcards.

A wildcard may be substituted for a defined subset of the $n$-grams.

Given a query $q$, retrieve all $n$-grams that match $q$. 
Wildcard N-Gram Retrieval

Given a set of $n$-grams, $n \leq 5$, and their frequencies. A query $q$ defines a pattern as a sequence of $n$-grams and wildcards. A wildcard may be substituted for a defined subset of the $n$-grams.

Given a query $q$, retrieve all $n$-grams that match $q$.

Straightforward solution:

- Construct a keyword index for the $n$-grams.
- Retrieve all $n$-grams that contain all of $q$’s words.
- Compile a pattern matcher from $q$ and filter the retrieved $n$-grams.

Improvements:

- Exploit information encoded in queries and $n$-grams, and that $n$ is small.
- Exploit closed retrieval settings, e.g., the $n$-gram set is constant.
- Trade wildcard expressiveness and retrieval recall for time.
- Exploit information about the application domain.
Wildcard N-Gram Retrieval

use the same ?

- Only 4-grams can match.
- First word use, second word the, third word same.

Our index stores information about \( n \)-gram length and word position in the pre-image of the index lookup function.

prefer * over

- 2- to 5-grams can match.
- First word prefer, and last word over.

Variable-length queries are sub-divided into fixed-length queries:
prefer over; prefer ? over; prefer ?? over; prefer ??? over

More search heuristics are described in [Stein et al., ECIR’2010]